het ontwerpen van interaktieve toepassingen en computernetwerken

handleiding voor strategisch en taktisch management, gebruikers en automatiseringsspecialisten

J.A. SCHELTENS

RAET
Software & Computer Services

ACADEMIC SERVICE

het ontwerpen van interaktieve toepassingen en computernetwerken

handleiding voor strategisch en taktisch management, gebruikers en automatiseringsspecialisten

J.A. SCHELTENS

ACADEMIC SERVICE

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Scheltens, J.A.

Het ontwerpen van interaktieve toepassingen en computernetwerken : handleiding voor strategisch en taktisch management, gebruikers en automatiseringsspecialisten / J.A. Scheltens ; [tek. F. Keyzer]. - Den Haag : Academic Service. - Ill., fig., tek.
Met lit. opg.
ISBN 90-6233-164-5
SISO 365.2 SVS 8.12.3 UDC 681.3.015:681.324 UGI 200
Trefw.: automatisering ; interactieve toepassingen / computernetwerken.

Uitgegeven door : Academic Service
Postbus 96996
2509 JJ 's-Gravenhage
Druk: Krips Repro Meppel
Bindwerk: Meeuwis, Amsterdam
Omslagontwerp: JAM Gauw
Tekeningen: F. Keyzer
ISBN90 6233 164 5

Proloog

Het boek

- De inhoud.

Zoals uit de titel blijkt gaat het om twee onderwerpen: interaktieve toepassingen en computernetwerken.
Bij het eerste onderwerp gaat het hoofdzakelijk om het ontwerpen van de interaktie tussen mens en computer via een beeldscherm. De benodigde software en de gegevens komen alleen in gebruikerstermen aan de orde voorzover dat voor de automatiseerder nodig is om de aansluiting met andere ontwerpaktiviteiten aan te geven. De kommunikatie tussen gebruikers en automatiseerders vormt de rode draad door het hele boek.
Het ontwerpen van computernetwerken is natuurlijk een heel ander, veel technischer onderwerp. Toch is het ontwerp van het hele systeem, inclusief het netwerk, gebaseerd op eisen van gebruikers. Het kwantificeren van gebruikers onder andere ten dienste van het konfiguratie- en netwerkontwerp vormt de tweede rode draad.
De inhoud zou ook kunnen worden weergegeven als in Fig. 1. Daarin zijn drie totaal verschillende werelden aangegeven met een aantal sleutelwoorden. Het boek is gebaseerd op de invulling van de twee witte vlekken die met vraagtekens zijn aangegeven. De pijlen geven aan dat er een stroom van informatie zou moeten vloeien. Helaas zijn de pijlen onderbroken!!

Aan allen die er onder
geleden hebben.

naar gebruikers en beide wijzen naar hun management. Daarnaast zijn soms managers gebruikers, informatie-analisten tegelijk systeemontwerpers en gebruikers bijna automatiseerders. Hoewel bij de splitsing in delen funktionarissen worden genoemd, gaat het om hun invalshoek. Dat betekent wel dat verschillende lezers verscheidene delen kunnen lezen.
Het strategisch management heeft meestal weinig tijd om zich met automatisering te bemoeien, daarom bestaat deel 1 maar uit weinig pagina's. Deel 2 is echter ook voor hen bedoeld!
Het taktisch gebruikersmanagement, zou na deel 2, ook nog deel 6 kunnen lezen, om beter met hun medewerkers te kunnen praten over de methoden. Voor de kommunikatie met hun kollega's in de automatisering zou het lezen van deel 3 goed zijn. Die kollega's zullen hen graag van dienst zijn als het te technisch wordt. Gebruikers die meer van de methoden willen weten en geinteresseerd zijn in de automatisering kunnen na deel 6, deel 4 lezen.
Het automatiseringsmanagement zou om de uitvoering van de methoden goed te managen na deel 3 nog deel 4 en 5 kunnen lezen.
Omdat alles in elkaar grijpt zijn de delen samengevoegd in één boek. De lezers die eigenlijk alles willen lezen maar helemaal geen tijd hebben kunnen dan nog terecht in de synoptische inhoud.
- Emancipatie.
In dit boek is niet gelet op de mannelijke of vrouwelijke persoonsvorm. In de omgeving van de schrijver werken vele vrouwen, die, naar zijn idee zo geëmancipeerd zijn, dat ze niet vallen over een woord als "manmaanden". Ze zien daarin juist een voorbeeld van gelijkwaardigheid.
- Voorkennis per deel.
Uiteraard is ieder deel geschreven voor een bepaalde groep lezers. Bij het strategisch management en bij gebruikers wordt geen echte automatiseringskennis verondersteld. Bij het taktisch gebruikersmanagement wordt die kennis ook niet verondersteld, maar we gaan wel in op een aantal details.
In de delen voor de automatiseerders wordt uiteraard de bij de funktie horende kennis aanwezig geacht. Bij transaktie-analisten betekent het, dat ze een kursus als (17) gevolgd moeten hebben, omdat we de daarin behandelde stof hier niet kunnen herhalen. Toch wordt de methode in voldoende mate behandeld om er mee te kunnen werken, vooral waar het om de resultaten gaat.

De schrijver

- Ervaring en kennis.
Alle problemen, vergissingen en fouten die in dit boek worden genoemd, zijn gebaseerd op de praktijk in allerlei soorten bedrijven. Het is daarom van belang een indruk te geven van de soort

bedrijven en de technische omgevingen, waarin de schrijver werkzaam is geweest.

Soort bedrijf	Technische omgeving
Onderdelen leverancier	Netwerk van minicoputers
Bezoekerscentra	Netwerk van minicomputers
Grondbeheer	Mainframe met terminalnetwerk
Bandenleverancier	Minicomputer
Bankinstelling	Mainframe met netwerk van minicomputers
Uitzendbureau	Minicomputer
Aannemingsmaatschappij	Terminals in service op mainframe
Instituut voor wetenschappelijk onderzoek	Minicomputer
Verkoopmaatschappij	Minicomputer
Verpakkingsindustrie	Netwerk van minicomputers
Valutahandel	Minicomputer
Inklaringskantoor	Netwerk van minicomputers
Supermarktketen	Mainframe met netwerk van minicomputers
Bankinstelling	Mainframe met netwerk van minicomputers
Sportorganisatie	Terminals in service op minicomputers
Grondbeheermaatschappij	Minicomputers
Leverancier van kantoormachines	Netwerk van minicomputers

Petrochemisch adviesbureau	Koppeling minicomputer-mainframe
Uitgeverij	Mainframe met terminalwerk
Distributiebedrijf	Minicomputer met terminalwerk
Bankinstelling	Mainframe, mini's, micro's
Papierfabriek	Minicomputers met procescomputers
Supermarktketen	Netwerk van minicomputers

De uitgevoerde opdrachten liepen uiteen van het houden van presentaties op direktie-niveau tot het opzetten van bedrijfskursussen voor eindgebruikers.
Aktiviteiten:
- het uitbrengen van adviezen over netwerkontwerp.
- het evalueren van de ergonomische aspekten van interaktieve toepassingen.
- het evalueren van de performance van computersystemen en netwerken.
- het houden van presentaties over netwerken, computeruitwijk, invoering van minicomputers, projektaanpak, methoden, ergonomie, kantoorautomatisering etc, op de verschillende niveau's:
  - strategisch management
  - taktisch gebruikersmanagement
  - taktisch automatiseringsmanagement
  - automatiseerders in de breedste zin van het woord
  - netwerkspecialisten
  - gebruikers
- het opzetten van specifieke kursussen en workshops voor gebruikers en automatiseerders

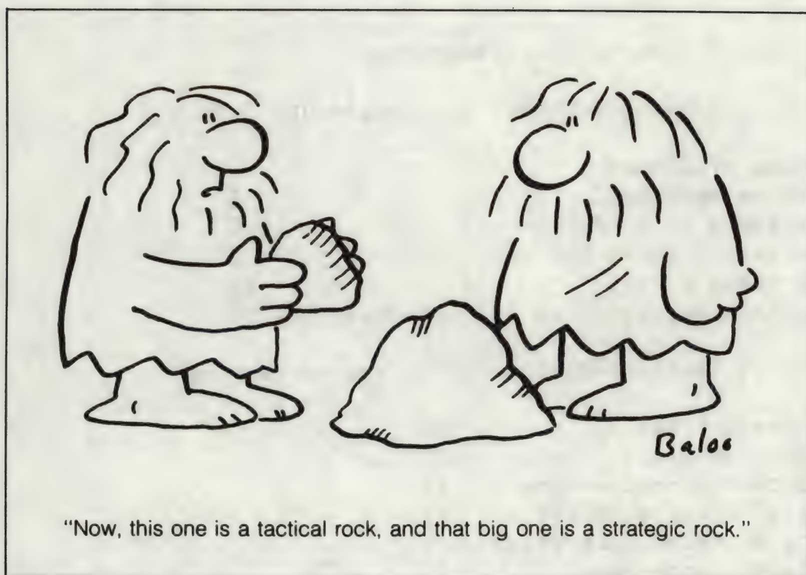
Er wordt regelmatig gesproken over "de praktijk". De uitspraken die gedaan worden zijn gebaseerd op de bovengenoemde praktijkervaring en de vele gesprekken met personen uit andere bedrijven tijdens lezingen, kursussen, seminars en andere bijeenkomsten, waarvan vooral de pauzes vaak zeer leerzaam waren!

- Erkentelijkheid.

Onbewust hebben in de afgelopen zes jaar, velen meegewerkt aan de tot standkoming van dit boek. Dat geldt niet alleen voor de vele mensen die ik heb mogen ontmoeten in allerlei bedrijven, maar ook voor verscheidene kollega's binnen RAET. Van die laatsten wil ik met name Herman Huis in't Veld en Nico de Gier noemen, die mijn kennis van databases en gegevensmodellen aanmerkelijk opgekrikt

hebben.
Een heel bijzondere plaats in de rij van onbewuste medewerkers wordt ingenomen door Hans Suys van Philips Telecommunikatie en Informatie Systemen B.V. in Den Haag. Tenslotte is hij de enige echte vader van de methode Transaktie analyse. Zijn handboek wordt nog steeds gebruikt in de kursus!
Vervolgens wil ik Addy Hensbergen dankzeggen voor het vele, vele typewerk dat ze voor me gedaan heeft, met grote opgewektheid en een geweldig uithoudingsvermogen! Mijn dank gaat daarbij tevens uit naar de andere dames van het sekretariaat, die er toch ook onder geleden hebben!
Toen de laatste loodjes van het schrijven van een boek zwaar begonnen te wegen en de planning van het tekenwerk onder druk kwam te staan, verklaarde Frank Keyzer zich bereid de tekeningen te maken. Bedankt Frank voor het tempo en de akkuratesse waarmee je gewerkt hebt!
Bijzonder dankbaar ben ik mijn zoon voor al het korrektiewerk dat hij verricht heeft en voor de vele adviezen ten aanzien van taal, stijl en woordgebruik. Als het boek ook nog enigzins leesbaar is geworden, is dat vooral aan hem te danken. Bedankt, Alex!
Tenslotte kom ik bij mijn vrouw. Ze heeft me niet aangemoedigd om aan dit boek te beginnen, ze heeft me anderhalf jaar lang zoveel mogelijk vrijgesteld van klusjes, bezoekjes en dergelijke en in de laatste fase heeft ze nog geholpen bij het korrektiewerk!
En daarmee ben ik in vele opzichten enige ervaringen rijker!

J.A. Scheltens
Apeldoorn 15-9-1985



"Now, this one is a tactical rock, and that big one is a strategic rock."

Inhoudsopgave

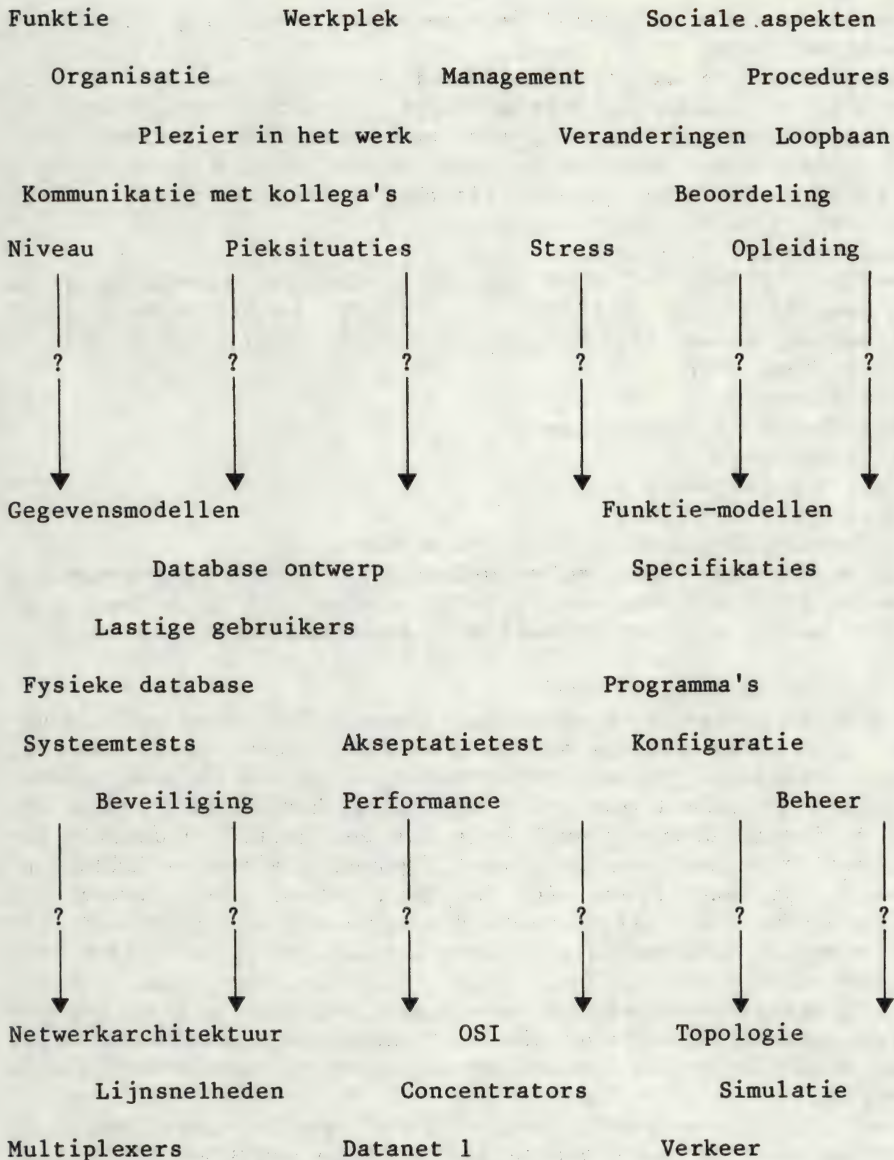


Fig. 1. Twee witte vlekken

De eerste witte vlek betreft de kommunikatie tussen gebruikers en automatiseerders.
De tweede witte vlek vormt het tweede onderwerp uit de titel: de kommunikatie tussen automatiseerders en netwerkontwerpers. De laatsten hebben cijfers nodig voor hun ontwerp, die automatiseerders zouden moeten verstrekken. Dat kunnen ze niet omdat tijdens het systeemontwerp niet gekwantificeerd wordt, zeker niet in het kader van de hoeveelheid verkeer.
- Waar het niet over gaat.
Hoewel zal blijken dat de te behandelen methoden aansluiten op al lang bekende projektaktiviteiten, zijn dat in dit boek randverschijnselen. Hoewel het verband ermee wordt aangegeven, is het niet een boek over:
- database-ontwerp
- distributed database-ontwerp
- programma-ontwerp
- keuze van pakketten
- vierde-generatietalen
- datakommunikatietechnieken en netwerken.
In de hoofdstukken over netwerkontwerp zal worden aangegeven wat de witte vlek precies inhoudt, maar het is niet de bedoeling de talloze boeken over datakommunikatie en netwerken te herschrijven.
- Een generaliserend verhaal.
Het boek is gebaseerd op ervaringen in vele bedrijven op alle niveau's. In ieder bedrijf was de situatie anders: andere mensen, andere strukturen, andere doelstellingen, andere achtergronden. Het geheel overziend komen toch een aantal hoofdlijnen steeds terug. Om die te beschrijven heb je dus een gemiddeld bedrijf nodig. Dan wordt het verschil tussen "generaliseren" en "middelen" minimaal. De lezer hoeft zich dus geen zorgen te maken want het gemiddelde bedrijf bestaat niet. Hij werkt bij een ander bedrijf, met een ander management, een ander beleid, een andere geschiedenis. Het effekt van het boek hangt af van de herkenning. Als die heeft plaatsgevonden kan de lezer aan het werk met de geboden aanpak. Aan het werk in de zin van: genuanceerd beoordelen welke aspekten van belang en toepasbaar zijn. Dat is helaas niet altijd hetzelfde.
De lezer wordt dus geacht twee vertaalslagen te maken. Eerst de vertaling naar de eigen bedrijfssituatie op basis van de herkenning van de problematiek. Vervolgens de vertaling van de geboden aanpak naar de eigen omgeving. De situatie van dat gemiddelde bedrijf, op de achtergrond van veel probleembeschrijvingen, wordt vaak zwart-wit beschreven in termen als "nooit", "altijd", "alle automatiseerders", "de gebruikers" etc. De lezer wordt verzocht

Hoofdstuk 42 Transakties

Hoofdstuk 43 Dialoogsimulatie

Hoofdstuk 44 Transaktie analyse

Deel 5, voor transaktie-analisten

Hoofdstuk 51 Mensen, methoden, middelen

Hoofdstuk 52 Transaktie analyse

Hoofdstuk 63 Dialoogsimulatie

Hoofdstuk 64 Transaktie analyse

DEEL 1
voor het strategisch management

Automatisering: de ivoren
toren van Babel.

Hoofdstuk 11
Mensen, methoden en management

11.1 Enige problemen

In een onbekend aantal bedrijven verloopt de automatisering geheel naar wens. Projekten zijn altijd stipt op tijd gereed, hebben niet meer gekost dan was overeengekomen, de gebruikers werken met plezier aan de beeldschermen, het ontworpen systeem sluit aan bij hun verwachtingen. Ten overvloede blijkt ook nog dat het aangeschafte systeem zeer flexibel is qua uitbreiding en kommunikatie met andere systemen, de exploitatiekosten lager zijn dan verwacht, kortom het geheel levert z'n geld op. De sociale gevolgen voor het personeel zijn per projekt van te voren in kaart gebracht, nieuwe taak/funktie-omschrijvingen zjn van te voren opgesteld en besproken met de ondernemingsraad en pas daarna is definitief besloten om het systeem te gaan bouwen.
Er zijn bedrijven waar het in sommige opzichten anders gaat. Bij bedrijven die een computer moeten aanschaffen, beginnen de problemen al op het moment dat de adviseurs van de direktie na enige jaren studie tot de konklusie komen dat voor het gehele bedrijf of voor een bepaalde vestiging maar eens aan een computer moet worden gedacht om bepaalde problemen op te lossen. Computerleveranciers bieden om strijdend hun produkten aan in een taal die niemand begrijpt. De folders zijn allemaal even aardig en sprekend, en de diskussie rond de kosten van het benodigde sys-

teem zijn niet meer te volgen. De leverancier spreekt in termen van geheugengrootte, aantal schermen, aantal schijven, MIPS, kilobytes, D.C.-poorten enzovoort. De gebruiker kan hoogstens de funkties aangeven waarbij de computer wordt ingezet.
Maar ook bij bedrijven waar men al jaren ervaring heeft met automatisering blijken er daarna nog meer problemen op te duiken. Hoe krijg je als strategisch management vat op de automatisering? Wat leek op de aanschaf van een stuk gereedschap voor het bedrijf, wordt een bedrijf binnen het bedrijf, dat goud kost zowel aan mensen als aan apparatuur. Er breken stakingen uit omdat er bij de invoering van beeldschermen ruzie ontstaat over de vraag welke funktionarissen nu eigenlijk met de beeldschermen moeten werken! Er blijkt geen weg terug te zijn. Bepaalde keuzes zijn onherroepelijk en zelfs niet meer te veranderen.
Verder blijkt op detailniveau, dat bijna ieder projekt

- meer kost dan oorspronkelijk werd geschat, toen op basis van de kosten/baten-analyse besloten werd door te gaan,
- langer duurt dan was geschat omdat automatiseerders te laat merkten wat de gebruiker eigenlijk bedoelde,
- niet het produkt oplevert dat de gebruikers verwachtten,
- organisatorische gevolgen heeft die niet waren voorzien,
- meer omscholing van de gebruikers eist dan was verwacht van gebruikersvriendelijke systemen,
- minder economisch resultaat oplevert dan was verwacht.

Het taktisch gebruikersmanagement vraagt zich af hoe automatiseringsprojekten kunnen worden beheerd qua tijd, kosten, voortgang, resultaat en kommunikatie met gebruikers. Het taktisch automatiseringsmanagement heeft problemen met opstellen van een informatieplan dat moet worden afgeleid van een niet bestaand bedrijfsbeleid. Het strategisch management waagt zich echter niet aan uitspraken die onkontroleerbare gevolgen hebben voor de automatisering. Daarmee is de cirkel gesloten: automatiseerders vragen om beleidsuitspraken om automatiseringsplannen op te baseren, het gebruikersmanagement doet die niet omdat ze niet weet wat daar de gevolgen van zijn. In de praktijk wordt de cirkel meestal doorbroken doordat de automatiseerders beslissingen nemen en hun plannen gaan uitvoeren. Gebruikers hullen zich in een veilig stilzwijgen en wachten af. Daarmee hebben ze het heft in handen gegeven van de automatiseerders.
Op uitvoerend niveau leidt dit wat de automatiseerder betreft tot uitspraken als: Gebruikers weten niet wat ze willen, doen nooit uitspraken, noemen geen cijfers en hebben nooit tijd.
Van de kant van de gebruikers worden opmerkingen gehoord als:

- "Automatiseerders vragen altijd naar zaken waar nog geen mens over heeft nagedacht".
- "Nu praten we al zo lang met ze en nog is het systeem niet

klaar".
- "Er is geen terugkoppeling: wat gebeurt er als....",
- "Al onze antwoorden op vragen van automatiseerders worden vastgelegd in een taal die voor ons onbegrijpelijk is".
- "Hoe kun je nu eisen stellen aan een produkt dat je niet kent"?
- "Als we zien of de automatiseerders ons goed begrepen hebben, is het te laat".

Met bovenstaande opsomming is niet alles gezegd, bovendien komen natuurlijk niet alle problemen altijd overal voor. Wie echter geen van deze problemen herkent, kent zijn bedrijf niet of heeft nog nooit met automatisering te maken gehad.

11.2 Waarom methoden

Laten we eens aannemen dat er een werkwijze zou bestaan, die, bij nauwkeurige navolging, alle genoemde problemen zou oplossen. Het eerste wat er dan moest gebeuren was die werkwijze nauwkeurig vastleggen, vervolgens gebruikers en automatiseerders voorschrijven op die manier te werk te gaan en tenslotte kontinu kontroleren of dat ook gebeurt. Karakteristiek voor een methode is dat het moet gaan om een uitgekristalliseerde, gestandaardiseerde en goed gedokumenteerde werkwijze. Als het gaat om een methode die een samenwerking tussen twee groepen tot stand moet brengen, moet door beide partijen voor de methode gekozen worden, op basis van de resultaten.

Het heeft geen zin een methode te kiezen waar kreatieve gebruikers zich helemaal in kunnen uitleven, maar waar automatiseerders niet verder mee kunnen. Het heeft evenmin zin een methode te kiezen waarbij automatiseerders met gebruikers kommuniceren via schema's, symbolen en termen die gebruikers niets zeggen.

Het gaat niet om twee partijen die tegenover elkaar staan, maar om twee soorten vakmanschap, waartussen een samenwerking moet ontstaan. De methoden moeten het vakmanschap van de automatiseerder, koppelen aan de materiedeskundigheid van de gebruikers en dat moet tot een resultaat leiden dat beter is dan de som der delen zou doen vermoeden.

Het moet bij methoden gaan om een uitgekristalliseerde werkwijze. Dat betekent dat het nut in de praktijk is bewezen en dat alle kinderziektes er uit zijn verdwenen. Methoden moeten ook overdraagbaar zijn. Er zou kursus in gegeven moeten kunnen worden. In zo'n kursus moet worden behandeld

- welke voorbereiding nodig is,
- uit welke stappen de methode bestaat,
- wat per stap moet gebeuren,
- welke dokumenten moeten worden ingevuld,
- welke resultaten worden opgeleverd,

- wat de relatie is met andere methoden en
- hoe de methode gemanaged moet worden.

In deze zin is het interviewen van gebruikers of het inschakelen van gebruikers bij het ontwerpproces geen methode. Iedere automatiseerder doet dat op zijn eigen manier en legt het resultaat vast op een wijze die hem goeddunkt.
Managers in de automatisering, die enige ervaring hebben met het invoeren van methoden, kennen de problemen die dat oplevert. Op strategisch niveau is er maar een ding van belang en dat is ervoor zorgen dat de kommunikatie tussen gebruikers en automatiseerders volgens beheersbare methoden wordt uitgevoerd. Er worden immers enorme bedragen uitgegeven aan automatisering die moet dienen om het bedrijf, en dat zijn alle gebruikers samen, beter te laten funktioneren. Het specificeren van de funkties en het afwegen van kosten tegen baten, kan dus niet in het ongewisse gelaten worden. Waar het gaat om toepassingen met beeldschermen zijn methoden beschikbaar die aan de gestelde eisen voldoen. Deze methoden zullen niet alle genoemde problemen doen verdwijnen, ze leveren wel een bijdrage in het oplossen van de meeste. Daarnaast moeten methoden worden uitgevoerd door mensen. Methoden in dienst van de kommunikatie tussen mensen met een geheel verschillende achtergrond, vragen wel om mensen die bereid zijn te kommuniceren. We zullen nu de genoemde problemen samenvatten en dan een aanpak bespreken.

11.3 Maatwerk of konfektie

Het ontwikkelen van een geautomatiseerd systeem is een zeer ingewikkeld gebeuren. Er zijn niet voor niets zoveel specialisten op een automatiseringsafdeling aanwezig. Het eindprodukt is een gereedschap dat gebruikt zal worden door niet-automatiseerders. Zij zitten achter beeldschermen, vragen informatie op, toetsen gegevens in, wijzigen gegevens enzovoort. Het is onmogelijk iets nieuws te ontwikkelen dat meteen goed is. Nieuwe dingen ontstaan langs een moeizame weg van proberen, proefmodellen maken, testen, verbeteren, weggooien, opniew beginnen enzovoort. De ontwikkeling is een herhaald, een iteratief proces.
Bij de keuze van konfektie-artikelen is het enige probleem de keuze. Het gaat dan om een kant en klaar produkt, dat door iedereen beoordeeld kan worden. Veel topmanagers denken bij automatisering aan het aanschaffen van iets. De computer wordt aangeschaft en dan moeten nog even, volgens specifikaties van de deskundigen, wat programma`s gebouwd worden. Maar die deskundigen onder de gebruikers moeten dan een situatie specificeren die ze nog niet kennen. Daarom zou het ontwerpen van geautomatiseerde systemen ook moeten verlopen via het boven omschreven proces van

proberen, aanpassen, opnieuw beginnen enzovoort. In de praktijk komt er van iteraties natuurlijk niets terecht alleen al vanwege de kosten. Er wordt opdracht gegeven een projekt uit te voeren. Dat kost al enorm veel moeite en blijkt achteraf altijd veel meer gekost te hebben dan verwacht werd en opnieuw beginnen is er niet bij. Dat betekent eigenlijk dat het eerste probeersel meteen het eindprodukt is. Als er erg veel klachten zijn, wordt er nog links en rechts wat aangepast, maar het systeem, dat toch al teveel heeft gekost, blijft zoals het is. De hele ontwikkeling had een iteratief proces moeten zijn, maar is een eenmalig proces geworden.
In feite ontstaat het hele probleem doordat gebruikers tijdens het ontwerp een situatie moeten specificeren die ze nog niet kennen. Een afdelingschef kan zich nooit precies voorstellen hoe de manier van werken wordt, als iedereen straks achter een beeldscherm zit, hoe de kommunikatie tussen medewerkers zal verlopen, welke dokumenten overbodig worden en hoe de transaktie aan het beeldscherm dus het beste kan worden ontworpen. De gemiddelde chef beschouwt het werken met beeldschermen als een gegeven, een produkt van de automatiseerders, waarop hij met zijn organisatie zo goed mogelijk moet inspelen. Automatiseerders verwachten echter van hem dat hij met zijn mensen specificeert hoe er met beeldschermen gewerkt moet worden.
De ontwikkeling van een automatiseringsprojekt verloopt in een aantal fasen.

- De analysefase, waarin de handmatige situatie wordt beschreven, en knelpunten en problemen naar voren komen. De huidige situatie wordt geanalyseerd in samenwerking tussen gebruikers en automatiseerders.
- De ontwerpfase, waarin een voorstel wordt uitgewerkt, meestal door de automatiseerders. Deze fase bestaat uit twee delen, het logisch en het technisch ontwerp. Bij het logisch ontwerp gaat het om een principe-oplossing: zo zou het met een beeldscherm kunnen. Bij het technisch ontwerp zijn de computer en de beeldschermen gekozen en wordt het principe-ontwerp uitgewerkt voor die bepaalde hardware.
- De bouwfase, waarin het systeem wordt gebouwd door de automatiseerders.
- De opleverings- en invoeringsfase, waarin het ontwikkelde systeem gaat funktioneren binnen het bedrijf. Als het goed is moet hier ook de evaluatie plaatsvinden. Vaak gebeurt dat niet omdat de mogelijkheden voor wijzigingen minimaal zijn en de automatiseerders allang met een ander projekt bezig zijn.

Voor de gebruikers is eigenlijk maar een trajekt belangrijk in dit hele gebeuren en dat is het logisch ontwerp. Als ze zich op dat moment het geautomatiseerde systeem volledig zouden kunnen

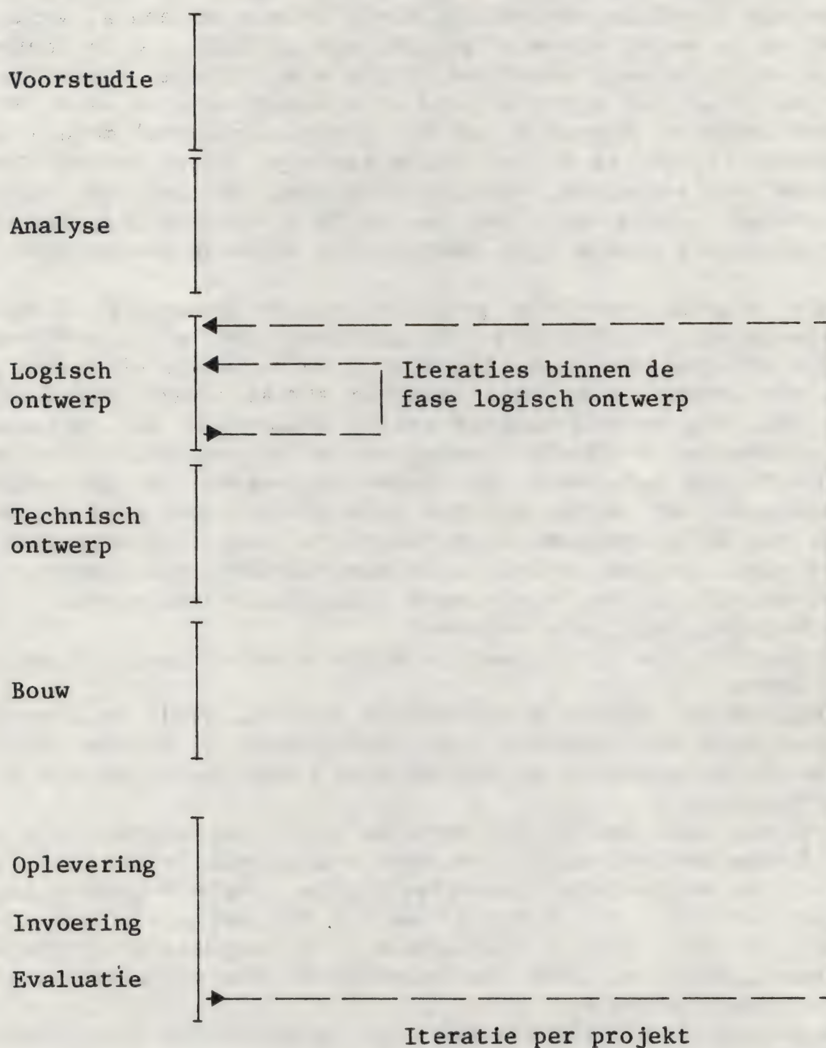


Fig. 11.1 De ontwikkeling van een systeem.

vormen van kommunikatie met gebruikers.
Bij een P1-projekt zal het vaak gaan om een C5N-omgeving. Men kan dan een globale logische Transaktie analyse uitvoeren, maar als het netwerk de bepalende faktor is in een kosten/baten-analyse, zou dialoogsimulatie moeten worden uitgevoerd om op basis daarvan een logische Transaktie analyse uit te kunnen voeren.
Bij een P2-projekt zal het meestal ook om een C5N-omgeving gaan. De logische Transaktie analyse levert dan de basis voor de topologie van het netwerk. De technische Transaktie analyse levert de cijfers voor de konstruktie en de apparatuur van het netwerk.
Voor P3-projekten in een CxN-omgeving, dus de verzameling van alle omgevingen waarin een netwerk voorkomt, leveren de logische en de technische Transaktie analyse de gegevens voor netwerkontwerp, responsetijden en systeembelasting. In C1- en C2-omgevingen kan men ter diskussie stellen hoe noodzakelijk het is de systeembelasting in kaart te brengen, als er al talloze toepassingen draaien. Daar moet op taktisch niveau over beslist worden. Het gaat om een rijdende trein: er draaien allerlei soorten toepassingen en niemand weet de systeembelasting van de diverse transakties. Als men echter nooit begint met het in kaart brengen van de belasting die transakties betekenen, blijft de situatie zo. In sommige gevallen is het gewoon een noodzaak om alsnog van een deel van de bestaande transakties de systeembelasting te bepalen. Denk aan uitwijksituaties, klachten over lange responsetijden en vervanging van systemen.
P4-projekten zullen zich niet meer afspelen in C1- of C2-omgevingen: bedrijven die qua grootte behoefte hebben aan een rekencentrum met een of meer mainframes hebben dat allang. Hoogstens gaat het om een uitbreiding, maar dat valt in het kader van dit onderwerp onder P3-projekten. Zowel in grote als in kleine bedrijven komen P4-projekten voor in een C3- of C3N-omgeving. In die gevallen gaat het om de aanschaf van een minicomputer. In veel gevallen zijn de aangeschafte minicomputers te klein, omdat het meestal om interaktieve toepassingen gaat, waarvan niemand de belasting kan aangeven. Computerleveranciers maken daar handig gebruik van: de konfiguratie wordt eenvoudig bepaald door de prijs en die hangt weer af van het aanbod van de konkurrenten. Daarom moet met het aanvragen van offertes gewacht worden tot het logisch ontwerp van de alle toepassingen gereed is die voorlopig op dat systeem voorzien zijn. De termijn speelt daarbij een belangrijke rol. Toepassingen die pas over drie of vier jaar zullen gaan draaien hoeven niet in het ontwerp te worden betrokken. Een konfiguratie eens per drie jaar uitbreiden is immers geen probleem. Het taktisch gebruikersmanagement wordt aangemoedigd om eerst logische ontwerpen in opdracht te geven. Dan zijn er voldoende ontwerpgegevens en cijfers van Transaktie analyse bekend om over de per-

bonden via een netwerk.

P6: Overgang van batch- naar on-line-systemen. Deze situatie komt het meest voor in mainframe-omgevingen C1 en C2.

P7: Evaluatie van bestaande systemen.

We zullen nu voor een aantal van de meest voorkomende kombinaties het gebruik van de methoden bespreken.

Het zal wel duidelijk zijn dat dialoogsimulatie overal wordt toegepast waar het gaat om het ontwerpen van interaktieve gegevensverwerkende toepassingen. In een P7-situatie is dat dus niet nodig en in C4-omgeving niet als het gaat om spreadsheets of zelfgebouwde toepassingen.

We zullen nu verder alleen nog letten op Transaktie analyse. De ergonomische Transaktie analyse levert gegevens op die voor de gebruikers van belang zijn, zoals aantal beeldschermen, aantal beeldschermuren per dag en de bezetting van de beeldschermen. Het gebeurt regelmatig dat noch gebruikers, noch automatiseerders zich de tijd gunnen voor dit soort ontwerpmethoden, terwijl achteraf vaak blijkt dat gebruikers heel andere of minder eisen hadden gesteld als ze van te voren hadden geweten hoe het systeem, wat hen betreft, zou werken. Het is dus van taktisch belang om per projekt, misschien per soort toepassing voor te schrijven welke methode zal worden toegepast. Daarbij moet het taktisch management natuurlijk weten wat het effekt is van methoden als Transaktie analyse. Een goede manier om dat te leren is het uitvoeren van een aantal proefprojekten.

Ergonomische Transaktie analyse is een manier om kwantitatieve gegevens van gebruikers, direkt om te rekenen naar konkekwenties en hen daarmee te konfronteren. Afhankelijk van het effekt van die cijfers blijken gebruikers daarna meestal veel meer geneigd hun cijfers nog eens kritisch te bezien. Cijfers in vergaderingen van stuurgroepen doen wonderen! Stuurgroepen zitten bijvoorbeeld vaak erg vast aan een eens bepaalde opleveringsdatum. Als uit cijfers van Transaktie analyse blijkt dat het aantal beeldschermen en dus het aantal medewerkers achter beeldschermen sterk afhangt van bepaalde cijfers van gebruikers en dat het nog veel tijd kost op verschillende vestigingen die cijfers nader te onderzoeken, dan moet de stuurgroep toch wel goed weten wat ze doet als ze ondanks deze wetenschap toch blijft vasthouden aan de vastgestelde datum. Een goed rapport en een goede presentatie, voorzien van cijfers en konklusies, kan men niet naast zich neerleggen. Zelfs in situaties waarin gebruikers helemaal geen cijfers willen of kunnen geven, zijn de resultaten van Transaktie analyse belangrijk. Daar kunnen de automatiseerders een aantal aannames doen en die omrekenen naar konsekwenties. Dan mogen de gebruikers hun eigen situatie kiezen. Een goede rapportage en presentatie spelen daarbij weer een belangrijke rol: ook dat zijn

nikatie tussen gebruikers en automatiseerders is meestal hetzelfde. In het midden- en kleinbedrijf maakt men alleen gemakkelijker de fout geen tijd en geld beschikbaar te stellen voor analyse en ontwerp. De direktie koopt een computer en die moet overmorgen werken, liefst met software uit de winkel op de hoek. De karakteristiek "grootte van de onderneming" heeft dus niet zoveel invloed op de toepassing van methoden.

- De grootte van de systemen en hun aantal. Het verschil tussen aanwezige of aan te schaffen systemen komt bij het soort projekten aan de orde. In de onderstaande tabel is een verdeling aangegeven die zinvol is voor de bepaling van het doel van de toepassing van de methoden.

Systemen	Geen netwerk	Netwerk	Doel van het netwerk
Een centraal rekencentrum met een of meer mainframes	C1	C1N	Remote terminals
Verscheidene rekencentra	C2	C2N	Remote terminals, koppeling tussen de centra
Een minicomputer	C3	C3N	Remote terminals
Een of meer microcomputers	C4	C4N	Micronetwerk
Kombinatie van mainframes, mini's, en/of micro's	C5	C5N	Remote terminals, computer-computer-koppelingen.

In grote bedrijven zullen we vaak de C1-, C2- of C5-omgeving aantreffen. In het midden- en kleinbedrijf komen we meestal C3, C4 of C5 tegen. Maar nogmaals, waterdicht is geen enkele indeling: in een groot bedrijf kan best een op zichzelf staande C4N-omgeving voorkomen, die in een klein bedrijf de hele automatisering aan zou kunnen.

- Het soort projekten gekoppeld aan het soort omgeving vormt een belangrijke sleutel voor de bepaling van het nut van methoden.

P1: Het opzetten van een automatiseringsplan met een globaal netwerkontwerp.

P2: Netwerkontwerp op basis van de gegevensdistributie.

P3: Nieuwe toepassingen op bestaande systemen.

P4: Nieuwe toepassingen op aan te schaffen systemen.

P5: Nieuwe toepassingen die voortvloeien uit de overgang van een

31.2 Methoden en omgevingen

We zullen eerst de methoden en hun resultaten in telegramstijl bespreken en vervolgens een aantal omgevingen bespreken. Tenslotte kunnen we dan het verband vaststellen tussen methoden en omgevingen.
Bij interaktieve toepassingen gaat het om het ontwerpen van de procedure rond het beedlscherm: de transaktie, en dus om allerlei menselijke handelingen en de dialoog met de computer via het beeldscherm.
Transaktie-ontwerp bestaat uit twee delen:
- dialoogsimulatie met als resultaat het kwalitatief ontwerp van transakties
- Transaktie analyse met als resultaat het kwantitatief ontwerp van de transakties:
  - gevolgen voor gebruikers,
  - systeembelasting en performance,
  - netwerkontwerp en netwerkbelasting,
  - basis voor performancebeheer.

Afhankelijk van de omgeving, de fase in de projektaanpak of de gewenste resultaten kan Transaktie analyse op drie manieren worden uitgevoerd.
- Ergonomisch: - gevolgen voor de gebruikers
- Logisch : - gevolgen voor de gebruikers
  - verwachte systeembelasting
  - verwachte extreme responsetijden
  - verwachte netwerkbelasting
- Technisch : - gevolgen voor de gebruikers
  - systeembelasting
  - responsetijden
  - netwerkbelasting

Uit de onderlinge vergelijking van de resultaten van de verschillende vormen van Transaktie analyse kan worden afgeleid dat het bij de analyses gaat om een toenemende detaillering van de technische aspekten. De logische analyse is een uitbreiding van de ergonomische, de technische een detaillering van de logische. De omgevingen kunnen worden ingedeeld op basis van een aantal karakteristieken
- De grootte van het bedrijf. In eerste instantie is er een groot verschil tussen enerzijds grote multinationals met vele automatiseringsafdelingen en allerhande specialisten en anderzijds het midden- en kleinbedrijf dat aarzelend overgaat tot de aanschaf van een minicomputer. Maar of een multinational nu een minicomputer aanschaft voor een bedrijfsonderdeel of een klein bedrijf doet hetzelfde voor de totale automatiseringsbehoefte, de kommu-

uitgedrukt is een beoordelingscijfer van 0 tot 10.
Of er methoden beschikbaar zijn voor de diverse aspekten binnen een projektaanpak, wordt eveneens beoordeeld met een cijfer van 0 tot 10. Wanneer er binnen de projektaanpak slechts aangegeven is dat er beeldschermen getekend moeten worden, dan is dat een nul voor methoden. Wanneer is aangegeven hoe beeldschermen worden gemaakt, op welk moment, als vervolg waarop, volgens welke standaards, door wie en met wie dan gaan we in de richting van de zeven. Wanneer die aanpak is vastgelegd, opgenomen in de training van automatiseerders en gebruikers, kompleet met de te gebruiken middelen, dan wordt het al een acht of misschien zelfs een negen.
Niet bij alle methoden passen gereedschappen. Per methode moet dus bekeken worden welke middelen er beschikbaar zijn. Daarna kan een methode beoordeeld worden op de beschikbaarheid van middelen.
Ook het management wordt qua vakmanschap beoordeeld in een cijfer van 0 tot 10. De projektaanpak zou kunnen worden aangegeven met ja of nee, wel aanwezig, niet aanwezig. Dan wordt met projektaanpak bedoeld de aanwezigheid van een aangekochte of zelf ontwikkelde projektaanpak. Bij dat zelf ontwikkelen zijn alle gradaties tussen "we doen maar wat" en komplete handboeken mogelijk. In dat geval dus toch maar een beoordelingscijfer. Een nul komt in de praktijk niet voor. Zelfs als ieder voor zich "maar wat doet", dan doet hij het nog op een bepaalde manier, bewust of onbewust.
De laagste beoordeling is hier een 1.
Opvallend is in de formule de plaats van M4: het management als exponent van de projektaanpak. De formule bestaat eigenlijk uit twee delen. Het ene deel bevat het vakmanschap de methoden en de middelen. In het andere deel gaat het om het kader waarin het eerste deel moet funktioneren: de projektaanpak en het management. Een getal tot de macht nul is nog altijd 1. Met andere woorden: zonder management, maar met goed vakmanschap, kunnen de resultaten redelijk zijn. Met een projektaanpak en deskundig management vliegen de resultaten echter omhoog. Wanneer het vakmanschap zou kunnen worden ingevuld met negens, maar het management met een nul, dan is de projektaanpak niet meer van invloed en wordt het resultaat 819. Wanneer daarnaast projektaanpak een acht krijgt en management een zeven, dan wordt het al 819 x 2097152.
De rest van het verhaal laat zich raden. Tenslotte is in de formule de betrekkelijkheid van methoden en gereedschappen aangegeven: het goede vakmanschap geeft een grotere invloed op het resultaat. Sommige automatiseerders werken het liefst zonder gestandariseerde methoden. Aan hun vakmanschap moet getwijfeld worden.
Het gaat hier uiteraard niet om een, in de praktijk bewezen formule. Wel wordt ermee aangegeven het belang van een aantal essentiele aspekten in de automatisering.

betekent niet dat hij alle details van alle methoden moet kennen, maar wel dat hij moet weten waarom er voor bepaalde methoden is gekozen. Het is van taktisch belang om te kontroleren of dit het geval is. De keuze van methoden is een verantwoordelijkheid van het taktisch management. Met de keuze is het echter nog niet af. Het uitvoerend management dient nog getraind te worden in de invoering van de gekozen methoden. Alle keuzekriteria, argumenten voor en tegen dienen te worden overgedragen. Voor- en nadelen dienen te worden besproken. Met name de korte-termijnnadelen dienen vergeleken te worden met de voordelen op langere termijn. Kortom, het uitvoerend management moet voorzien worden van alle middelen om de toepassing van methoden te verkopen en te blijven verkopen. Van de 4 M`s is de laatste zeker niet de minst belangrijke. De praktijk bewijst het tegendeel. Hoe goed de methoden ook zijn, zonder effektief management is het resultaat nihil.
Men zou deze paragraaf in de vorm van een formule kunnen samenvatten.

$$PR = M1\ (1 + M2\ (1 + M3))\ PA^{M4}$$

PR: Het projektresultaat
M1: Het vakmanschap
M2: De methoden
M3: De middelen
M4: Het management
PA: De projektaanpak

De resultaatformule

Het projektresultaat kan natuurlijk worden bezien vanuit diverse gezichtspunten: de planning, de kosten, de performance, de tevredenheid van de gebruikers.
Zonder methoden, middelen en management kunnen enkele zeer gemotiveerde, zeer deskundige vakmensen best een systeem bouwen met goede resultaten. De grap is dat iedere automatiseerder meestal denkt dat hij tot die groep behoort. In de praktijk is bijna niemand zo gemotiveerd, dat hij zonder management kan werken. De deskundigheid valt nogal eens tegen. Veel jongeren moeten nog een groot stuk bedrijfservaring opdoen. Het vakmanschap van veel automatiseerders beperkt zich tot een deel van de automatisering, terwijl er systemen voor gebruikers gebouwd moeten worden. Bovendien kan niemand geacht worden zijn leven lang bij een bedrijf te blijven werken en deskundigheid is dus nogal vluchtig. Tenslotte is het zo, dat er vaak een groep automatiseerders samenwerkt en dan is het werken volgens een en dezelfde methode van levensbelang voor het welslagen van een projekt. Het vakmanschap wordt

keuze van methoden is het uiteraard van belang vast te stellen hoe ze op elkaar aansluiten en passen binnen een gekozen projekt-aanpak. In de paragrafen over projektaanpak wordt daar uiteraard dieper op ingegaan.
De derde M is terwille van de alliteratie van "middelen". Het had ook "gereedschappen" of "tools" mogen heten. Het is prettig om het eens te zijn over een methode als dialoogsimulatie, maar zonder simulator gaat dat niet zo eenvoudig. Evenzo is het voor Transaktie analyse handig om over het rekenprogramma te beschikken omdat het handmatig rekenwerk niet zo eenvoudig is. De dialoogsimulator kan een beeldscherm zijn aan een willekeurige computer maar in de praktijk blijkt een microcomputer die portable is, het handigst te zijn. Dat geldt zelfs voor bedrijven met vele computers en vele beeldschermen. Het gaat immers vaak juist om die gebruikers die nog niet werken met beeldschermen. Wanneer er meerdere computers in het bedrijf aanwezig zijn, is het voor de informatie-analist handig om steeds met hetzelfde gereedschap te werken.
De laatste van de 4 M's is die van het management. Hoewel er aan het uitvoerende management natuurlijk vele kanten zitten, gaat het in dit verband om de invoering van methoden. Het liefst werkt iedere automatiseerder volgens zijn eigen methode, of het nu gaat om flow charts, programmastrukturen of entiteitsmodellen. Waar men mee is opgegroeid, daar houdt men aan vast. Nu is er natuurlijk wat voor te zeggen om te werken met de methode waarmee men de meeste ervaring heeft. Binnen een automatiseringsafdeling is dat niet akseptabel. Aan het ene bureau worden Bachmandiagrammen getekend, aan het andere JSP struktuurdiagrammen en aan het volgende flow charts. De kwaliteit van het geheel is belangrijker dan de kwaliteit van het produkt aan een bepaald bureau. De argumenten tegen het overschakelen op andere methoden blijken vaak even hardnekkig als ridikuul. Bijna altijd komt het neer op korte-termijn-denken. Omschakeling kost nu te veel tijd, brengt de doorlooptijd van het projekt in gevaar, levert voor dit projekt geen voordeel op. Nog vreemder wordt het bij de invoering van nieuwe methoden. Dan is er geen alternatief, maar toch worden de meest oneigenlijke argumenten aangevoerd om maar op de bekende natte-vinger-manier te kunnen blijven werken.
Kortom, het invoeren van methoden van werken kost energie en vereist vakmanschap. Het management zou de noodzaak van toepassing van konsistente methoden moeten inzien. Dat kan alleen met een management dat voldoende van het vak af weet. Veel uitvoerende managers worden door de eerste de beste ondergeschikte onderuit gehaald wanneer het gaat om voor- en nadelen van bepaalde methoden. De medewerker denkt daarbij aan zijn eigen werk, de argumenten van de manager zouden van een ander gehalte moeten zijn. Dat

kan verschillen. Van ervaring leren we immers wel eens iets. Als dat ergens voor geldt, is het wel voor de kommunikatie met de gebruiker. Iedere situatie is anders, iedere gebruiker is weer anders, denkt anders over automatisering. Er zijn dan ook geen informatie-analisten die niet kommuniceren met gebruikers en er zijn bijna geen bedrijven, waar gebruikers zich niet afvragen wie dit allemaal heeft bedacht en wie er nu eigenlijk om deze manier van werken gevraagd heeft. Daarom is het van essentieel belang methoden in te voeren voor de kommunikatie met gebruikers. Methoden die hun waarde in de praktijk hebben bewezen en die voorkomen dat we elders gemaakte fouten allemaal nog eens maken. Methoden die in dat kader van belang zijn, zijn Transaktie analyse en dialoogsimulatie. In andere paragrafen wordt ingegaan op deze methoden zelf. Nu stellen we alleen vast dat in het kader van de kommunikatie met de gebruiker vakmanschap alleen niet voldoende is. Wanneer in een projekt die uitstekende informatie-analisten volgens hun eigen methode kommuniceren met gebruikers, dan kost dat geld, het levert verwarring bij de gebruikers en wanorde in de dokumentatie. Er wordt wel eens opgemerkt dat methoden de kreativiteit doden. Men bedoelt gewoon: "Ik wil op mijn eigen manier werken". Dat soort kreativiteit kunnen we ons echter niet permitteren. Het is veel konstruktiever kreatief te zijn in het ontwerpen van bijvoorbeeld een overzichtelijke schermlay-out of van een handige alternatief voor de dialoog. Automatiseerders lijken vaak op konstrukteurs die zich druk maken over de aanzichten die er van een kontruktie getekend moeten worden, en met welke soort pen, in plaats van te letten op de bruikbaarheid en de bediening. Methoden betekenen een zeer bepaalde manier van werken. Kreativiteit ligt op een ander terrein. Het dialoogontwerp is meestal toonbeeld van fantasieloosheid van de ontwerpers. Daar was kreativiteit op z`n plaats geweest. Zelfs het maken van schilderijen gaat volgens methoden.
In het kader van dit boek gaat het om de methoden Transaktie analyse en dialoogsimulatie. Transaktie analyse is een begrotingsmethode. Deze analysemethode start met het maken van een transaktieschema in het kader van het transaktie-ontwerp. Dat transaktieschema wordt beoordeeld en aangepast via dialoogsimulatie. Dan volgt de vertaling naar het detailschema en vervolgens wordt het rekenproces gestart dat een hoeveelheid resultaten oplevert die verder geanalyseerd en verwerkt dienen te worden.
Transaktie analyse levert enerzijds gegevens op die voor de gebruiker van belang zijn, anderzijds gegevens die voor het netwerkontwerp noodzakelijk zijn. Zoals alle begrotingsmethoden kan ook Transaktie analyse gebruikt worden in verschillende graden van nauwkeurigheid. Tijdens het logisch ontwerp zal de nauwkeurigheid anders liggen dan tijdens het technisch ontwerp. Bij de

werken. Maar dit terzijde. Het ging over het vakmanschap van de dialoogontwerper. In de praktijk zal dat de informatie-analist zijn. Wanneer hij met de gebruiker de transaktieschema's heeft opgesteld behoort het tot zijn vakmanschap dat hij een van mogelijke dialoogvormen kiest die het beste past bij wat de gebruiker wil. Daar is wat meer kennis voor nodig dan weten wat menuschermen zijn. Het is ook niet voldoende om klakkeloos te doen wat de gebruiker voorstelt. Een gebruiker mag inderdaad uiteindelijk zelf bepalen hoe hij wil werken, maar pas nadat hij alternatieven heeft gezien. Ervaren gebruikers kunnen trouwens heel goed vaststellen hoe vakkundig informatie-analisten zijn op dit gebied. Wanneer er nooit een tegenvoorstel komt of een alternatief voor een betere oplossing, beginnen de wenkbrauwen omhoog te gaan.
Een ander soort vakmanschap is gelegen op het gebied van de omgang met de gebruiker. Het zich instellen op de problemen van de gebruiker, op zijn denkwereld, zijn terminologie, zijn manier van werken. Op dat terrein zijn er weinig kursussen en boeken, maar het is van groot belang voor het taktisch management om zich er van te overtuigen dat of het ook echt goed gebeurt. Het is niet voldoende om vast te stellen dat er met gebruikers is gepraat, of dat er transaktieschema's zijn gemaakt, of dat er dialoogsimulatie is uitgevoerd. Het gaat erom vast te stellen of methoden en middelen gefunktioneerd hebben zoals ze bedoeld zijn.
Een derde soort vakmanschap heeft te maken met de sociale aspekten van de automatisering. In hoeverre heeft de informatie-analist gevoel voor problemen van de gebruiker, die niets met beeldscherm of dialoogontwerp te maken hebben? Het gaat er uiteindelijk immers om, een projekt tot een goed einde te brengen. Een projekt is geslaagd, wanneer gebruikers tevreden, misschien zelfs enthousiast zijn over de nieuwe manier van werken. Wanneer gebruikers niet tevreden blijken te zijn en blijven klagen over het werken met beeldschermen dan is het uitermate vervelend dat de dialoog ergonomisch uitstekend en zeer gebruikersvriendelijk is, maar de gebruiker gewoon gevoelsmatige of bedrijfspolitieke problemen heeft met de automatisering op zijn bureau. Het zou niet de eerste keer zijn dat een gebruiker vindt dat zijn werk niet te automatiseren is. Wanneer de informatie-analist geen antenne heeft voor de signalen van die gebruiker, dan mislukt het projekt hoe prachtig de schermlay-out en hoe flitsend de responsetijden ook zijn!
De tweede M is van methoden. Er wordt wel eens gezegd dat er geen eigenwijzer volk bestaat dan automatiseerders en misschien is dat wel zo. In ieder geval is het zo dat ze eigenlijk maar een methode de beste achten en dat is hun eigen methode. Vaak gaat het helemaal niet om een methode maar om een manier van werken die wordt aangepast aan de situatie en die van projekt tot projekt

Hoofdstuk 31
Mensen, methoden, middelen

31.1 Vier × M

Mensen, methoden, middelen en management. Met mensen wordt in dit geval bedoeld hun vakmanschap. Er zijn verbazingwekkend veel boeken geschreven over ergonomie in verband met beeldschermen. Even verbazingwekkend is echter wat er op het gemiddelde beeldscherm in het gemiddelde bedrijf staat. Wat dat betreft lijkt het wel of er geen boeken of kursussen bestaan. Wanneer ontwerpers gewezen wordt op de gebrekkige, onhandige, gebruikersonvriendelijke dialoog, geven ze meestal reakties in de stijl van: "Niet genoeg tijd gehad" of "De gebruikers zijn best tevreden" of "Het wordt nog wel veranderd". Het lijkt wel alsof de kennis over dialoogontwerp moeilijk in praktijk te brengen valt. In de praktijk blijkt vaak, dat wat een ontwerper ziet als eerste poging of voorstel, een eigen leven gaat leiden en heel snel een status krijgt die het eigenlijk niet toekomt. De oorzaak daarvan is weer dat een dialoogvoorstel natuurlijk gebouwd is. Er zijn schermlayouts gemaakt en programma's geschreven. Veranderingen daarin op grond van ergonomische overwegingen worden meestal nooit doorgevoerd. Het werkt immers. Er is altijd wel een gebruiker te vinden die er enthousiast over is. Vooral als hij de altenatieven niet kent. Het kost meestal niet veel moeite om ervoor te zorgen dat hij die nooit ziet. Maar alleen al deze gang van zaken die in de praktijk veel voorkomt is al een reden om met dialoogsimulatie te

DEEL 3

voor het taktisch automatiserings management

Er is behoefte aan nog een methode: die om methoden in te voeren.

lang dat de uiteindelijke computer daarbij niet nodig is.
Omdat de toekomstige situatie kwalitatief en kwantitatief is na het logisch ontwerp, kan dan al begonnen worden met de voorbereiding van de invoering.

gegevens onbruikbaar is om goede responsetijden te halen. Ook bij dit soort projekten is transaktie-ontwerp onmisbaar.
Samenvattend kunnen we vaststellen dat de meeste fouten voortkomen uit een te geringe of verkeerd gerichte betrokkenheid bij de automatisering.

23.5 Adviezen

Automatisering is een zaak van het bedrijf, de gebruikers en de automatiseerders. Het topmanagement van het bedrijf besluit tot automatisering over te gaan op basis van ekonomische argumenten. Gebruikers moeten met het systeem werken en dienen begeleid te worden. De automatiseerders moeten het ontwerp maken en het systeem bouwen. Op alle drie de terreinen bestaan genoeg specialisten, maar de bruggen ertussen ontbreken. Wat is het nut van allerlei bijeenkomsten waar agogische trainers praten over het bedrijf, de organisatiestruktuur, de hierarchieke verhouding, de werksfeer enzovoort, als de gebruiker daarna, teruggekeerd achter zijn bureau, zich probeert voor te stellen wat dat nu allemaal te maken heeft met de komende beeldschermen. De resultaten van dit soort sessies zijn vluchtig. De tijd kan beter besteed worden aan het opleiden van gebruikers tot medeontwerpers, tot gebruikers die een toetsenbord aandurven raken, zinnige eisen stellen en met nieuwe voorstellen komen.
Het gebruiken van methoden houdt volgens de definitie van methoden in, dat er een standaardmanier van werken is. Dat betekent dat er voor of door gebruikers een planning gemaakt kan worden van de benodigde tijd. Een projekt moet in een aantal stappen in opdracht worden gegeven. De eerste stap loopt uiteraard tot en met het vooronderzoek. Op basis van de grote lijnen wordt opdracht gegeven het logisch ontwerp te maken met als eerste aktiviteit het maken van een planning voor automatiseerders en gebruikers. Aan het eind van het logisch ontwerp is het te bouwen systeem al door de gebruikers beoordeeld. Dit is voor gebruikers en automatiseerders de belangrijkste fase. Alle andere fasen zijn ervan afgeleid. Deze fase mag dan ook de meeste tijd kosten. Vervolgens wordt het technisch ontwerp in opdracht gegeven. Als daarvan de planning is ingediend weten de gebruikers wat het moment is om te kontroleren of inderdaad aan alle gestelde eisen is voldaan. Aan het eind van het technisch ontwerp moeten de automatiseerders een realistische opleveringsdatum kunnen afgeven. Die datum mag worden rondgebazuind in de gebruikersorganisatie.
Er moet besloten worden te kiezen voor methoden voor de samenwerking tussen gebruikers en automatiseerders, waarbij gebruikers konkreet achter een beeldscherm hun nieuwe manier van werken helpen ontwerpen. Bij nieuwbouw op nieuwe hardware is het van be-

Een andere fout is de gedachte dat er veel automatiseringskennis nodig is om de automatisering te managen. Het enige wat taktische gebruikers aan automatisering moeten doen is toezien op de selektie van methoden die de samenwerking tussen gebruikers en automatiseerders tot stand brengen. Die methoden moeten in de eerste plaats aansluiten bij het kennisniveau van de gebruikers en in de tweede plaats aansluiten op methoden die automatiseerders toepassen bij de bouw van het systeem.
Het is fout te geloven dat voor een logisch ontwerp de computer al moet zijn aangeschaft. Het logisch ontwerp is een ontwerp op papier en op een simulator, maar onafhankelijk van het uiteindelijke systeem. In omgevingen waar een computer moet worden aangeschaft, kan er dus heel veel gebeuren zonder dat er een computer is gekocht.
Het is fout te denken dat systeemontwikkelingsmethoden of een projektaanpak er alleen is voor de automatiseerders. Gebruikers moeten precies weten wat hen te doen staat in de analysefase, tijdens het logisch en tijdens het technisch ontwerp.
Het is fout te denken dat gebruikers inspraak mogen hebben in het ontwerp van het systeem, nee, zij behoren mede-ontwerpers te zijn. Dat heeft niets te maken met allerlei agogische trainingen, maar met methoden die hout snijden.
Een andere veel voorkomende fout bij first time users is een ongenuanceerd enthousiasme. Vooral managers die zelf initiatief genomen hebben om te gaan automatiseren maken deze fout. Ze nemen adviseurs in dienst die meestal meer verstand hebben van organiseren dan van automatiseren en denken dat er niets meer fout kan gaan. De adviseurs trekken een jaar of meer uit voor interviews, rapportage, voorstellen enzovoort, en tenslotte moet er nog een bureautje gezocht worden dat even de programma's maakt. Vervolgens komen langzaam maar onontkoombaar de tekorten, de onnauwkeurigheden en de onduidelijkheden in de specifikaties boven water.
Een fout die regelmatig gemaakt wordt is het inschakelen van gebruikers van het type C en het te laat of helemaal niet inschakelen van de mensen die uiteindelijk met de beeldschermen moeten werken. Soms is type C zo enthousiast dat ze denken alles het beste te kunnen. Het is alleen om psychologische redenen al fout om het type A en/of B niet te betrekken bij het transaktie-ontwerp.
Nog steeds wordt de fout gemaakt dat een bestaand batch-systeem even omgezet kan worden naar een interaktief systeem. Het toetsenbord vervangt de ponskaart en het scherm de printer en verder hoeft er niets te veranderen. Zo wordt het dus een kort en goedkoop projekt. Niets is minder waar. Bij evaluatie van dit soort projekten blijkt dan ook altijd dat het koncept niet deugt, de programmatuur onvoldoende is aangepast en dat de toegang tot de

dus transaktieschema's en ergonomische resultaten van Transaktie analyse.
Transaktieschema's moeten worden gemaakt om de gehele procedure aan het beeldscherm te beschrijven, schermlay-outs moeten worden gekontroleerd op leesbaarheid, schermindeling en informatie per scherm, transaktiedefinities dienen om te kontroleren of en in hoeverre dialoogsimulatie is uitgevoerd en de ergonomische resultaten van Transaktie analyse tonen aan dat die analyses zijn uitgevoerd.
Wanneer een standaardpakket wordt gekocht ligt de zaak in principe eenvoudig omdat er altijd een systeem te vinden is waarop het pakket getest kan worden. Het interaktieve aspekt van het pakket kan door de betrokken gebruikers beoordeeld worden. Om ook de andere aspekten van een pakket te kunnen beoordelen moet eerst een kompleet logisch ontwerp gemaakt worden alsof men het systeem ging bouwen. Vervolgens moeten de funkties en de gegevensstrukturen van het ontwerp vergeleken worden met die van het pakket. Dan is ook gemakkelijk in kaart te brengen wat er gewijzigd of toegevoegd moet worden. Een pakket kopen, uitsluitend op basis van specifikaties van de leverancier is vragen om problemen.
Samenvattend kunnen we zeggen dat het managen van de automatisering neerkomt op het managen van de ontwerpfase. Het op hoop van zegen voorbij laten gaan van deze fase kan zomaar een jaar vertraging opleveren bij de invoering. Heel triest wordt het, als de automatiseerders weinig tijd hebben kunnen besteden aan inbreng van gebruikers omdat in de stuurgroep het gebruikersmanagement een niet-realistische opleveringsdatum heeft gekozen.

23.4 Taktische fouten

De grootste en meest voorkomende fout is dat men in de gebruikerswereld denkt, dat automatisering een zaak is van automatiseerders. Hoogstens vindt men dat de materiedeskundigen aan automatiseerders moeten uitleggen hoe er nu gewerkt wordt, maar wat er straks aan de beeldschermen moet gebeuren, moeten de automatiseerders maar bedenken.
Een direkt gevolg van deze eerste fout is een tweede, waar automatiseerders bijna dagelijks mee te maken hebben: er wordt onvoldoende tijd beschikbaar gesteld voor gebruikers in projektgroepen en werkgroepen. Het eigen, eigenlijke werk heeft altijd een hogere prioriteit.
De volgende, hieruit af te leiden fout is het prikken van een opleveringsdatum aan het eind van het vooronderzoek, het rondbazuinen van die datum in de organisatie en het eraan vasthouden, koste wat kost. Het is voor automatiseerders vaak zeer interessant eens uit te zoeken waar de opleveringsdatum op gebaseerd is!

schermlay-outs op papier of op een beeldscherm. Maar werken met een beeldscherm is nu eenmaal iets anders dan een schermlay-out beoordelen. Meestal legt de informatie-analist geduldig uit wat de beste oplossing is. Te grote afwijkingen van de plannen van de automatiseerders zullen zeker het projekt vertragen.

- Het beheer van de voortgang. De meeste gebruikers hebben geen idee van de fasen in een projektaanpak. Automatiseerders illustreren dit door te wijzen op gebruikers die tijdens de bouw nog met voorstellen komen. Veel gebruikers kennen maar twee hoofdpunten in een projekt: de interviews met informatie-analisten en de oplevering. Soms worden na de interviews nog scherm- en printlay-outs besproken.

De besproken methoden zullen nooit alle problemen oplossen, maar ten aanzien van de bovengenoemde zal het volgende gebeuren.

Dialoogsimulatie geeft inzicht in de kwalitatieve aspekten van de automatisering en maakt een zeer goede kommunikatie tussen gebruikers en automatiseerders mogelijk.

Transaktie analyse levert gegevens voor de kwantitatieve aspekten van de automatisering. Transaktie-ontwerp schept bij gebruikers een duidelijk gevoel voor de afsluiting van hun inbreng en geeft eventueel het startsignaal voor iteraties. De technische resultaten van Transaktie analyse maken aan het eind van het technisch ontwerp een laatste afweging van eisen tegen resultaten mogelijk.

- Centraal ontwikkelde of gekochte pakketten. Bij grote bedrijven komt het regelmatig voor dat voor een aantal vestigingen, die niet beschikken over een automatiseringsafdeling, centraal een pakket wordt ontwikkeld.

Voor de automatiseerders ontstaat nu het probleem dat ze niet meer weten wie <u>de</u> gebruiker is. Dit kan tot wanstaltige ontwerpen leiden. In een poging die onbekende gebruiker toch tevreden te stellen, wordt een hoeveelheid flexibiliteit in het systeem gebouwd die prachtig lijkt, maar vaak zoveel overhead kost dat de performance slecht wordt.

Voor de gebruikers kan het probleem ontstaan dat procedures die per vestiging verschillen, uniform moeten worden tengevolge van de automatisering. In principe hoeft dat niet slecht te zijn. Als de verschillen niet wezenlijk zijn is het zonde allerlei versies van hetzelfde systeem te bouwen en te onderhouden. Gebruikers mogen zich dus echt wel eens af vragen hoe belangrijk bepaalde verschillen nu eigenlijk zijn.

Voor het management van de vestigingen is er maar een manier om de automatisering in dit soort situaties te managen en dat is het eisen van de toepassing van overeengekomen methoden voor de kommunikatie. Men behoort zoveel van die methoden of te weten dat precies bekend is wat de gebruikers moeten doen en welke dokumenten er opgeleverd moeten worden. Bij transaktie-ontwerp zijn dat

tiseerders aan het ontwerp willen beginnen mogen de gebruikers eisen stellen. Voor veel gebruikers is dat net zoiets als een oerwoudbewoner vragen naar eisen die hij aan een cityhopper zou stellen. Het resultaat is dan ook vaak dat gebruikers het automatiseren maar aan de vakmensen overlaten en wachten op de oplevering van het produkt. Ze zijn geneigd zich niet te verdiepen in hun aandeel in de automatisering, geen uitspraken te doen, geen cijfers te noemen en bovenal, geen tijd te hebben. "Zelfs als men start vanuit de beste bedoelingen, blijven systeemontwerpers, gebruikers en management veelal met een kater zitten" (23). Daarbij realiseren ze zich niet dat, wat in een fabriek het proefmodel zou zijn, in de automatisering meteen het eindprodukt is. Er is geen alternatief en geen budget om een ander produkt te maken: in de automatisering bestaat geen weg terug. De visie van de automatiseerders op dit probleem wordt vaak samengevat in de opmerking: gebruikers weten niet wat ze willen.

- Onvoldoende inzicht in het eindresultaat. Op het moment dat de opdracht een projekt uit te voeren bestaat, is er geen enkel inzicht in kwalitatieve noch de kwantitatieve aspekten van de eindsituatie. Hoe zal de akseptatie door de eindgebruikers zijn? Wat zal de performance zijn? Wat gaat het geheel kosten? Wie leest de ontwerpdossiers? Hoe gemakkelijk is het werken met beeldschermen? Hoe verandert de funktie-inhoud?

In de kwantitatieve aspekten bestaat meestal het minste inzicht. Hoeveel mensen zitten er straks achter beeldschermen? Hoeveel beeldschermuren maken ze per dag? Hoeveel uren moeten er gemaakt worden in pieksituaties? Hoeveel werk blijft er liggen tot de volgende dag? Hoe is de werkverdeling over de beeldschermen? Kan een bepaalde hoeveelheid werk voor een zeker tijdstip zijn uitgevoerd? Hoe lang worden de wachtrijen voor het loket?

Funktioneel gezien krijgen de automatiseerders een blanko cheque. Bijna alle projekten lopen uit en welke keus hebben gebruikers als ze gekonfronteerd worden met vertraging en extra kosten?

Het ontwerpen van komplexe systemen is altijd een iteratief proces. Zelfs de meest ervaren gebruikers hebben moeite om allerlei haken en ogen te voorzien. In de automatisering betekent de oplevering het begin van de eerste iteratie. Er zijn gebruikers die al jaren wachten op de resultaten. Het iteratieve proces moet plaatsvinden tijdens de ontwerpfase van een projekt.

- De kommunikatie met de automatiseerders.

Informatie-analisten die de bestaande situatie in kaart moeten brengen, gaan daarbij te werk met een nauwgezetheid, gedetailleerdheid en een logica, die de meeste gebruikers overrompelt. Vervolgens wordt de bestaande situatie vastgelegd in schema's die geen gebruiker kan lezen, tenzij hij tot het type D behoort. Daarom wordt hem de toekomstige situatie getoond in de vorm van

Een derde variant is de micro met verscheidene beeldschermen. Toen de minicomputer werd ingevoerd dachten veel autoatiseerders dat de kapaciteit van een grote computer nu in een iets kleinere verpakking werd aangeboden. Talloze overbelaste mini's in allerlei bedrijven bewijzen deze overschatting. De geschiedenis herhaalt zich. Een micro die de bestandsbenaderingen van een aantal andere micro's moet uitvoeren, raakt natuurlijk een keer overbelast. En weer zal niemand weten wanneer, totdat het gebeurt.
Ten aanzien van de performance van microsystemen en het gebruik van micro's voor terminal emulatie geldt dat methoden als Transaktie analyse even hard nodig zijn als bij beeldschermen aan minicomputers of mainframes.
Bij kantoorautomatisering denken we aan geintegreerde funkties ten dienste van kantoorpersoneel op werkplek-, afdelings- en concernniveau. Kantoorautomatiseringsfunkties worden zelden gebouwd door een automatiseringsafdeling. Een tekstverwerker wordt niet ontwikkeld, maar gekocht.
Voor kantoorautomatiseringsfunkties zijn dus meestal geen ontwerpmethoden nodig. Voorlopig zullen we echter nog vele jaren allerlei gegevensverwerkende funkties ontwikkelen en ook die maken deel uit van de kantoorautomatisering. Ten aanzien van het netwerk dat de kommunikatie mogelijk moet maken ligt de zaak anders omdat dat wel ontworpen moet worden. Dat betekent dat van de gebruiker wel cijfers worden verwacht over de frequentie waarmee hij de verschillende funkties zal gebruiken. Ook hier geldt dat het ontwerpproces een iteratief proces is waarbij gebruikers naar prijskaartjes mogen en moeten vragen voor alternatieve oplossingen die afhangen van verschillende schattingen van de frequenties. Kortom, het heeft enige kleine voordelen om bottum-up, op grote schaal, ongestruktureerd micro's in te zetten, maar dan moet wel parallel daaraan top-down, een kantoorautomatiseringsplan met een kommuniaktie-infrastruktuur ontstaan.

23.3 Taktische problemen

Alle problemen die gebruikers hebben met de automatisering kunnen worden samengevat onder de noemer: het managen van de automatisering. In deze samenvatting kunnen we vier aspekten onderscheiden.
- Problemen rond het stellen van eisen.

Wanneer er na allerlei voorstudies projekten zijn gedefinieerd, gaan de automatiseerders volgens een vast basispatroon aan het werk: analyse, ontwerp, bouw en oplevering. Hoewel de analysefase het gemakkelijkst voor de gebruiker zou moeten zijn, blijken zich op detailniveau vaak al zoveel problemen voor te doen dat veel gebruikers dan de draad al kwijtraken en nauwelijks kunnen kontroleren of de automatiseerders het goed doen. Wanneer de automa-

kunnen bijdragen in dit proces.
Wanneer we alleen bestaande situaties automatiseren is het bijna zeker dat we als bedrijf het computersysteem niet volledig benutten.

23.2 Microcomputers en kantoorautomatisering

In het kader van de reeds jaren bekende automatisering met grote centrale computers, zou men het gebruik van microcomputers de maximale decentralisatie kunnen noemen. Er kunnen allerlei argumenten worden aangevoerd voor decentralisatie. Een eigen minicomputer op een kantoor geeft een stukje onafhankelijkheid van het hoofdkantoor. De microcomputer biedt de gebruiker de mogelijkheid zijn persoonlijke visie op automatisering snel te realiseren zonder allerlei moeilijke diskussies met automatiseerders, die hem toch niet begrijpen en jaren nodig hebben om te bouwen wat hij niet nodig heeft. Gebruikers die zich enigzins zouden verdiepen in het funktioneren van een automatiseringsafdeling zouden tot de konklusie komen dat daar een groot aantal specialisten samenwerkt: informatie-analisten, systeemontwerpers, programmeurs, operators, beheerders en onderhoudspersoneel. Geen wonder dat alles zo traag gaat en zoveel geld kost. De gebruiker gaat het zelf doen. Op grote schaal worden door bedrijven micro's aangeschaft en uitgedeeld aan de zich verdringende gebruikersmenigte. Na de spelletjesperiode begint men aan het serieuze werk. En of ze nu met BASIC of met spreadsheets gaan werken, langzaam maar zeker begint het tot hen door te dringen dat ze hun problemen moeten analyseren, ontwerpen moeten maken, programma's moeten maken en intoetsen, hun computer moeten leren bedienen, een administratie moeten opzetten om alles te beheren en regelmatig hun programma's moeten aanpassen aan de nieuwste wensen. Kortom, in iedere micro-eigenaar zijn alle automatiseringsfunkties terug te vinden. Geen wonder dat veel gebruikers toch weer afhaken. De doorzetters lopen vroeg of laat aan tegen het probleem van de beperkte kapaciteit en het gebrek aan kommunikatie met grotere computers met grotere bestanden of databases. De microcomputer mag dan de funkties hebben, meestal zitten de benodigde gegevens in de grote computer. Om toch iets aan de beperkingen te doen zijn wat oneigenlijke oplossingen bedacht. Met terminal-emulatie kan de microcomputer fungeren als beeldscherm aan een grote computer. Daarmee is dan meteen een degradatie optreden, want van de eigenschappen van de microcomputer is niets meer over. Het is een dom beeldscherm geworden, bestuurd door een interaktieve toepassing op de grote computer. Een andere oplossing is gevonden in het opzetten van een netwerk van microcomputers. Een van die micro's fungeert als een soort gegevensbeheerder: de fileserver.

- per afdeling : het tijdbestedingsdiagram,
- per werkplek : de uit te voeren transakties,
- per transaktie: het transaktieschema.

Dat betekent dat personeelszaken nauwgezet in de gaten moet houden wanneer de werkplekken met beeldschermen worden gedefinieerd en transakties per werkplek worden vastgesteld en ontworpen.

- De creatieve gebruiker.

Laten we nog eens even in het kort herhalen hoe automatisering in z'n werk ging, voor zover het de gebruiker betrof. De informatie-analist voerde een analyse uit van de bestaande administratieve procedure. Een goede analist was natuurlijk ook op de hoogte van administratieve en boekhoudkundige zaken. Hij had al snel door hoe de zaak werkte, welke gegevens waar nodig waren, wat er mee gebeurde en welk rekenwerk werd uitgevoerd. Op basis van die analyse maakte hij een ontwerp voor een informatieverwerkend systeem, al dan niet met beeldschermen uitgerust. Kortom, hij automatiseerde de bestaande situatie. Dat daardoor het werk van bepaalde funktionarissen ineenschrompelde tot een stukje programmatuur was zijn zorg niet. Het was wel de zorg van de werknemers, maar wat zouden die er aan moeten doen, om over personeelszaken maar niet te spreken. Toch was het iedereen duidelijk dat het hier om zeer harde sociale aspekten van de automatisering ging. Nog steeds wordt er door talloze informatie-analisten op deze wijze gewerkt. De bestaande situatie wordt in kaart gebracht, op grond daarvan wordt een ontwerp gemaakt. Kortom, de bestaande situatie wordt geautomatiseerd. Daarbij kunnen bijna alleen maar banen wegvallen. Een ander bekend verschijnsel is dat gebruikers van beeldschermen na enige tijd met allerlei voorstellen komen voor nieuwe toepassingen. Het kan dan bijvoorbeeld gaan om informatieverstrekking aan een klant die in de handmatige situatie niet mogelijk was. Dat soort "achteraf voorstellen" zijn vaak aardig, maar meestal niet meer in te voeren omdat bijvoorbeeld de benodigde gegevens niet in de bestanden zijn opgenomen.

Een ander verschijnsel dat in dit verband van belang is, is het feit dat computers steeds meer worden toegepast op werkterreinen waar de gemiddelde informatie-analist niet zo goed thuis is. Nog steeds zal de computer in eerste instantie worden gebruikt voor de bekende administratieve zaken maar de andere afdelingen komen zeker aan de beurt. Waar zouden voorstellen van nieuwe aktiviteiten beter kunnen ontstaan dan in de gebruikerswereld zelf? Daar waar gebruikers mede-ontwerpers zijn geworden en via dialoogsimulatie hun nog niet gebouwde toepassingen uitproberen, is de situatie rijp om iets nieuws te bedenken. Op dat moment moet de gebruiker gemotiveerd worden om met nieuwe funkties te komen die werk kunnen betekenen. Gebruikers die er de tijd en de gelegenheid voor krijgen, blijken zeer inventief en zeer effektief te

transaktie zal uitvoeren moet geen nauwkeurig inzicht verwachten in de toekomstige geautomatiseerde situatie. Wanneer het van belang is om na te denken over de konsekwenties van het langdurig werken met beeldschermen dan is het van belang dat soort cijfers vast te stellen. Al zou het maar zijn op de manier van: minimaal, gemiddeld, maximaal. Het is natuurlijk te gek dat gebruikers inzicht willen hebben in hun toekomstige situatie, maar zelf geen getallen aangeven. Dan zou de onduidelijkheid en de ongrijpbaarheid van de automatisering in ieder geval niet aan de automatiseerders te wijten zijn.
Het gaat om de kombinatie van twee vakgebieden: dat van de gebruiker en dat van de automatiseerder. Wanneer de automatiseerder methoden toepast om de gebruikerssituatie kwantitatief in kaart te brengen, zal de gebruiker zijn aandeel moeten leveren in de vorm van aantal transakties per dag, pieken en dergelijke.
Wanneer dat gebeurd is en het tijdbestedingsdiagram is gemaakt, kan er konkreet gediskussieerd worden over het aantal beeldschermen per afdeling, het aantal uren per werkplek, een eventuele werkverdeling over de diverse werkplekken van een afdeling, opvang van pieksituaties enzovoort. Het tijdbestedingsdiagram is een waardevol dokument voor personeelszaken. Tezamen met de transaktieschema's vormt het een hulpmiddel bij het vaststellen van de nieuwe taak/funktie-omschrijving.
Beide dokumenten zijn immers leesbaar voor gebruikers. Wanneer dat niet het geval is zijn ze niet gemaakt zoals in dit boek is aangegeven in de delen voor de informatie-analisten en gebruikers.

- De nieuwe taak/funktie-omschrijving.

In veel bedrijven bestaat de taak/funktie-omschrijving niet. Er kan wel worden aangegeven wat de funktie zo ongeveer inhoudt, maar vastgelegd is het niet. In de bedrijven waar ze wel beschikbaar zijn, zijn ze meestal achteraf vastgesteld, vaak na diskussies van jaren. Bij de invoering van automatisering krijgt de taak/funktie-omschrijving een veel zwaarder aksent. Immers, wanneer men vanuit personeelszaken of vanuit de ondernemingsraad nog konkreet wil bijsturen dan zal de taak/funktie-omschrijving er eerder moeten zijn dan de automatisering. Bijsturen kan alleen in de ontwerpfase, en niet wanneer het systeem gebouwd is. Dat wil zeggen, wat er in bestaande systemen nog aan te passen valt, hangt af van het gebouwde systeem en niet van de wensen van de gebruiker. In het algemeen gaat het dan om marginale wijzigingen en niet om fundamentele verbeteringen. Een konkreet middel om een stem te krijgen in de automatisering is voor personeelszaken het opstellen van de taak-funktie-omschrijving tijdens de ontwerpfase.
Uiteindelijk moeten beschikbaar zijn:

ene gebruiker vriendelijk is, is voor de andere irritant. Bij de ene transaktie is een duidelijke begeleiding door het systeem noodzakelijk, bij de andere transaktie staat het werktempo voorop. Voor een nieuwe medewerker ligt de situatie vaak anders dan voor een ervaren personeelslid.
Kortom, gebruikersvriendelijkheid is per persoon en per transaktie anders. Nu zal het zeker niet zo worden dat er per medewerker een aparte dialoog met de computer ontworpen zal worden. Waar het om gaat, is dat medewerkers van personeelszaken en leden van de ondernemingsraad van tevoren willen kunnen beoordelen of dat nodig is. Vervolgens wil men van tevoren een inzicht hebben in de tijd die medewerkers straks doorbrengen achter hun beeldscherm. De methoden die in dit boek worden behandeld, leveren een konkrete bijdrage in de diskussie over de sociale aspekten van het werken met beeldschermen. Konkrete simulatie van de nieuwe funktieinhoud, konkrete cijfers over het aantal uren per dag. Dialoogsimulatie maakt het mogelijk om een werkplek in te richten met een beeldscherm en daarmee alle toekomstige transakties uit te voeren alsof de computer al beschikbaar is. Transaktie analyse op basis van deze gesimuleerde transakties levert per transaktie de terminaltransaktietijd (T.T.T.) op. Dat wil zeggen: het gemiddeld aantal seconden en de standaardafwijking per transaktie. Daarin is dan meegenomen dat binnen een transaktie bepaalde zaken soms wel en soms niet moeten worden ingetypt. Tevens is op het transaktieschema de aansluiting met de handmatige verwerking aangegeven. De T.T.T. is de tijd die nodig is om de transaktie uit te voeren, zoals die is vastgelegd op het transaktieschema.
Nu zullen er op een bepaalde werkplek vaak verschillende transakties worden uitgevoerd. Wanneer alle transakties via Transaktie analyse zijn gekwantificeerd en alle T.T.T.`s zijn bepaald kan een tijdbestedingsdiagram worden gemaakt. Daaruit blijkt de tijd die per werkplek besteed wordt aan de diverse transakties. Het aantal terminaluren per werkplek is het aantal uren dat een medewerker per dag achter het beeldscherm door moet brengen om alle transakties af te handelen. Het aantal terminaluren per transaktie is gelijk aan het aantal transakties per dag X de T.T.T. van die transaktie /3600. Het aantal terminaluren per werkplek is de som van alle terminaluren per transaktie voor die werkplek. De resultaten van dit rekenwerk worden door de automatiseerders vastgelegd op het tijdbestedingsdiagram.
Wanneer het aantal terminaluren per werkplek de acht overschrijdt, is er kennelijk iets aan de hand met de ontworpen transakties of met het aantal transakties dat door gebruikers is genoemd. Uiteraard zijn de gegevens van de gebruikers over de aantallen transakties per dag van groot belang. Een gebruiker die niet weet of hij een keer of honderd keer per dag een bepaalde

Hoofdstuk 23
Diversen

23.1 Sociale aspekten

- Sociale aspekten in cijfers

Er wordt reeds jaren gediskussieerd over het werken met beeldschermen. Met name over de nadelen die het kan hebben voor de gezondheid, zowel lichamelijk als geestelijk. Bij de geestelijke aspekten gaat het dan om een ondefinieerbare, onvoorspelbare graad van vermoeidheid die optreedt wanneer enige tijd intensief met een beeldscherm is gewerkt.
Ook hier is weer het probleem dat achteraf praten vrij eenvoudig is, terwijl de mensen van personeelszaken graag inzicht willen hebben in de geautomatiseerde situatie voordat die ontstaan is. Zoals bekend is er in de praktijk bij automatisering geen weg meer terug. Vandaar dat door vele deskundigen in de sociale sektor de automatisering gezien wordt als een ongrijpbaar en onafwendbaar gebeuren. Men heeft ztting in stuurgroepen en waarschuwt voor de vele onzekerheden ten aanzien van de invoering van de automatisering. De reakties zijn meestal even ongrijpbaar als de automatisering zelf. Termen als gebruikersvriendelijkheid, ergonomie, geleidelijke invoering, acceptatietest, begeleiding van gebruikers, klinken immers ontwapenend. Wat kan er eigenlijk nog fout gaan als systemen op die wijze worden ontworpen en ingevoerd? Wel, er kan nog steeds van alles fout gaan. Gebruikersvriendelijkheid bestaat niet, in z'n algemeenheid. Wat voor de

ma`s aanpassen voor de verwerking door de computer en het transport door het netwerk. Aan het eind van het technisch ontwerp zijn alle gegevens van de nog te bouwen situatie bekend en kan een goede schatting worden gegeven van responsetijden.
Andere resultaten van Transaktie analyse maken het mogelijk de belasting van alle beeldschermen op de computer te bepalen. In veel gevallen hebben lange responsetijden te maken met overbelaste computers. En daarmee zijn we bij een van de basisproblemen van interaktieve toepassingen. Niemand weet welke belasting allerlei transakties voor een computer betekenen. In de meeste bedrijven is verder niet in kaart gebracht op welke beeldschermen welke transakties worden uitgevoerd. Er is een konfiguratie besteld op advies van de leverancier en men is met de bouw van interaktieve toepassingen begonnen. Met de groei van het aantal transakties groeide ook het aantal beeldschermen en voor dat men wist wat er gebeurde was het systeem overbelast. In het algemeen zit er minstens een jaar tussen de eerste klachten over lange responsetijden en de uitbreiding van de konfiguratie. Als Transaktie analyse konsekwent wordt uitgevoerd, is tijdens het technisch ontwerp al bekend hoeveel belasting een projekt toevoegt aan de bestaande situaties.
Transaktie analyse is voor een deel gebaseerd op cijfers van gebruikers. Het is dus uiteindelijk in alle opzichten voor de gebruikers zelf van belang de meest realistische cijfers te verschaffen.

men zich in stilzwijgen hult en de automatiseringsafdeling het maar uit laat zoeken, dan blijven in de bedrijven systemen komen met onakseptabele responsetijden. En dat is dan nog maar een aspekt van het werken met interaktieve systemen!
Bij dialoogsimulatie hebben we al gezien dat de oplossing van het probleem van responsetijden begint bij het stellen van realistische eisen. Als gebruikers daar al geen tijd voor hebben, moeten ze niet verwachten dat er ooit wat zal veranderen op dit gebied. Direkt na dialoogsimulatie kan Transaktie analyse worden uitgevoerd voor de ontworpen transakties. Bij transaktie analyse wordt ook de verwerking door de computer, zoals die in de rechterkolom van het transaktieschema per interaktie kwalitatief is beschreven, kwantitatief in kaart gebracht. De automatiseerders geven dan op het detailschema voor iedere interaktie aan hoeveel I/O`s er gedaan moeten worden. Tijdens het logisch ontwerp is dat aantal slechts globaal aan te geven, tijdens het technisch ontwerp zijn de gegevens redelijk nauwkeurig. Het rekenprogramma berekent op basis van deze gegevens responsetijden. Tijdens het logisch ontwerp kunnen twee soorten Transaktie analyse worden uitgevoerd: de ergonomische en de logische, tijdens het technisch ontwerp wordt de technische analyse gerealiseerd.
In het algemeen vormen de responsetijden van een transaktie maar een klein deel van de T.T.T. Met andere woorden, berekening van het aantal beeldschermen en het aantal beeldscherm-uren wordt meestal nauwelijks beinvloed door responsetijden. Om de resultaten van de ergonomische Transaktie analyse te verkrijgen, kan het detailschema zodanig worden opgezet dat de verwerking niet wordt gespecificeerd, maar dat de gewenste responsetijden worden aangegeven. In dat geval kunnen de ergonomische resultaten worden vastgesteld en leiden tot konklusies voor gebruikers, ervan uitgaand dat aan de gestelde eisen zal worden voldaan.
Indien echter de gegevensanalisten de logische struktuur van gegevens al hebben ontworpen, is het ook mogelijk op het detailschema meteen de verwerking te specificeren op basis van die informatie. Nu kan er een groot verschil bestaan tussen een logische gegevensstruktuur en de uiteindelijke technische opslag in het geheugen van de computer. De responsetijden zoals die door het rekenprogramma worden bepaald zijn nog zeer onnauwkeurig. Het gaat erom dat automatiseerders gedwongen zijn de eisen van gebruikers konsekwent in kaart te brengen op het detailschema en dat extreem lange responsetijden vroegtijdig worden gesignaleerd en teruggekoppeld naar de gebruikers.
De logische Transaktie analyse levert alleen verwachte technische resultaten op. Wanneer er echter geen alarmerende situaties ontstaan, wachten de gebruikers het einde van het technisch ontwerp af. Tijdens die fase kunnen transaktie-analisten de detailsche-

voerregel worden verplaatst met enkele besturingstekens, het volgende moment wordt een scherm met een tekst van 2000 tekens overgestuurd. Een computer wordt aangeschaft, een netwerk is meestal maatwerk. De kosten van het netwerk worden onder andere bepaald door de benodigde kapaciteit. Door de methoden in dit boek behandeld, worden alle transakties in kaart gebracht en vertaald naar hoeveelheid verkeer. Dat betekent dus dat er in feite een vertaalslag uitgevoerd wordt van gebruikerseisen en gebruikersgegevens naar de benodigde netwerkkapaciteit. Daarmee is voor de gebruikers het plaatje van de responsetijden kompleet.
Samengevat komt het er dus op neer, dat er methoden zijn om onprettige verrassingen met responsetijden te voorkomen. Die methoden gaan echter wel uit van gebruikersgegevens. Zowel de aanschaf van een computer als het opbouwen van een netwerk zijn zeer kostbare zaken. Eenmaal gebouwd, zijn ze niet even aan te passen. Niet van de ene dag op de andere, zelfs niet van de ene maand op de andere. Dat betekent, nogmaals gezegd, dat het niet akseptabel is dat gebruikers zich geheel of gedeeltelijk onttrekken aan het mede-ontwerperschap. Er moet vanaf het begin een onherroepelijk beslag worden gelegd op de tijd van desbetreffende medewerkers. Het is van taktisch belang er op toe te zien, dat methoden worden geselekteerd die het mede-ontwerperschap doen funktioneren, dat die methoden konsekwent worden toegepast en dat er tijd beschikbaar is om de methoden in praktijk te brengen. De tijd dat gebruikers bij irritant lange responsetijden simpel met de vinger naar de automatiseringsafdeling kunnen wijzen raakt voorbij. Bovendien is daarmee niemand gebaat. Beter is het om via gerichte methoden gegevens te verstrekken die gebruikt kunnen worden voor dimensionering van konfiguraties en netwerken. Natuurlijk zullen er ten aanzien van sommige transakties zijn onzekerheden blijven bestaan. Dat mag echter nooit een reden zijn om dan maar niets te doen. Immers, overdimensionering is mogelijk, maar dan wel alleen, wanneer de 100% situatie goed bekend is. Er zijn al genoeg zogenaamd overgedimensioneerde systemen halverwege het projekt te klein gebleken. Alsof de technische aspekten nog niet komplex genoeg zijn, speelt bij de aanschaf van een computer natuurlijk ook de konkurrentie tussen de computerleveranciers een rol. Enerzijds zullen ze graag systemen leveren die groot genoeg zijn om flitsende responsetijden mogelijk te maken, anderzijds dringt de angst om een opdracht te missen hen de kant op van goedkopere en dus kleinere systemen. Die diskussies kunnen het beste gevoerd worden door de automatiseerders. De cijfers die voor die gesprekken nodig zijn zijn echter een afgeleide van de gebruikersgegevens. Dus kan tenslotte nogmaals worden vastgesteld dat de gebruiker moet bestellen wat hij nodig heeft. Gebrekkige, onvolledige en verkeerde cijfers leiden tot verkeerde systemen. Wanneer

werkt met de gesimuleerde transaktie en de relatie met handmatige aktiviteiten duidelijk wordt, blijkt heel snel wat de kritische interakties zijn en wat niet. Van de kritische interakties kan worden vastgesteld wat de maximale responsetijd is, door op de simulator een steeds langere responsetijd in te stellen. Dat betekent dat aan het eind van de simulatiefase exakt bekend is welke eisen er aan de responsetijden van de diverse interakties worden gesteld. Op het transaktieschema is van iedere interaktie aangegeven wat de verwerking inhoudt. Met andere woorden: in een zeer vroeg stadium wordt aan de automatiseerders heel nauwkeurig duidelijk gemaakt wat een bepaalde verwerking door de computer aan tijd mag kosten. Tijdens het technisch ontwerp moet worden vastgesteld in hoeverre dat haalbaar is. Immers, dan is het computersysteem, alsmede de struktuur van bestanden en databases bekend. In situaties waar de computer het reeds lang aanwezige mainframe is en waarin gebruik gemaakt moet worden van reeds bestaande bestanden of databases kan, mede op grond van aanwezige ervaring, zeer snel worden vastgesteld of de responsetijdeisen haalbaar zijn. Zo niet, dan mag de gebruiker alsnog zeggen of hij het zinvol acht de transaktie te realiseren. In deze situatie worden dan op de dialoogsimulator direkt de haalbaar geachte responsetijden ingesteld, en de gebruiker beoordeelt die.
Alsof dit alles nog niet komplex genoeg is, voegen we nog een tweede element toe aan het begrip responsetijd. Iedere interaktie betekent verwerking door de computer, zoals we zagen. Er wordt iets ingetypt, de computer zet iets op het scherm. Dat betekent dat de ingetypte tekens moeten worden getransporteerd naar de computer, die ze verwerkt, wat zal resulteren in een reaktie op het scherm. Die reaktie bestaat eveneens uit een aantal tekens dat getransporteerd moet worden van computer naar beeldscherm. Voor dat transport heen en terug is tijd nodig: hoe meer tekens, hoe meer tijd. Sommige beeldschermen hebben van de computer veel meer tekens nodig dan de tekens die op het scherm verschijnen. Een en ander is afhankelijk van de intelligentie die is ingebouwd in een beeldscherm. Hoe dommer het beeldscherm, hoe meer besturingstekens er nodig zijn om bijvoorbeeld een stuk tekst met een bepaalde lay-out te displayen.
We hebben gezien dat een computer een beperkte kapaciteit bezit in de vorm van het maximum aantal I/O-operaties per tijdseenheid. Ook een netwerk heeft zijn grenzen. Die liggen bij het maximum aantal te transporteren tekens per tijdseenheid. Hoe dichter die maximum kapaciteit benaderd wordt hoe langer het duurt om een bericht van een bepaald aantal tekens te transporteren. Het verkeersaanbod in tekens per tijdseenheid hangt af van het aantal aktieve beeldschermen en de lengte van de berichten. Op het ene moment moet alleen de cursor op het scherm naar de volgende in-

het aantal I/O-operaties per interaktie 20 zou bedragen, dan vormt de groep van beeldschermen een belasting van 1800 I/O-operaties per minuut. Iedere computer kent een maximum aantal uit te voeren I/O-operaties per tijdseenheid. Hoe dichter die grens benaderd wordt, hoe langer de responsetijden. Er moet dus een computersysteem gekozen worden dat krachtig genoeg is om een gegeven belasting aan te kunnen. Wanneer de in dit boek behandelde methoden worden toegepast, wordt het mogelijk om bijvoorbeeld het aantal beeldschermen, het aantal interakties per tijdseenheid en het aantal interakties per transaktie te berekenen. Dat betekent in feite echter niets meer, dan dat er een vertaalslag wordt gemaakt van gebruikerseisen en gebruikersgegevens naar bovengenoemde cijfers. Anders gezegd, er wordt een computer geselecteerd op basis van gebruikerseisen en -gegevens. Daarom is het van zo groot belang dat er in de gebruikersorganisatie voldoende tijd wordt vrijgemaakt om mee te werken aan het ontwerpproces. Wanneer het aantal transakties per dag onzorgvuldig wordt geanalyseerd, kan dat tot gevolg hebben dat bij de configuratiebepaling uitgegaan wordt van te weinig beeldschermen. Het gevolg daarvan is een te kleine computer. Het kan ook zijn dat de bezettingsgraad van de beeldschermen in de praktijk veel zwaarder is dan berekend werd op basis van de onnauwkeurige gegevens van de gebruikers. Daardoor kunnen er gedurende meer uren per dag meer beeldschermen in gebruik zijn dan verwacht. Een overbelaste computer levert door lange responsetijden een onvoorstelbare hoeveelheid ergenis en stof tot kankeren.
Afgezien van overbelasting kan een responsetijd uiteraard ook te lang worden omdat het aantal I/O-operaties voor een bepaalde interaktie erg groot is. Dat kan veroorzaakt worden door een te groot aantal bestanden dat benaderd moet worden. Beperking van het aantal bestanden moet dan overwogen worden door de automatiseerders, maar mogelijk is dat niet altijd. Uiteindelijk kan met behulp van dialoogsimulatie de responsetijd "hard" gemaakt worden voor de gebruikers en dan is aan hen de keus om al dan niet over te gaan tot de bouw van deze bepaalde transaktie.
Dat brengt ons terug bij het onderwerp gemiddelde responsetijd. Zoals aangegeven, is een gemiddelde responsetijd een onmogelijk begrip. In de eerste plaats omdat spreken over een gemiddelde alleen zinvol is wanneer ook de spreiding wordt opgegeven, in de tweede plaats omdat aangegeven moet worden over welke nog niet ontworpen transakties het gemiddelde wordt genomen en onder welke nog niet geanalyseerde omstandigheden en in de derde plaats omdat de gebruiker niet te maken krijgt met gemiddelde responsetijden, maar met konkrete responsetijden waarbij "lang" zeer betrekkelijk is. Bij een goede dialoogsimulator moet het mogelijk zijn per interaktie een responsetijd in te stellen. Wanneer de gebruiker

nen van het eerste teken van de reaktie van de computer op het scherm. Het intypen van gegevens loopt dus uit op een interaktie met de computer. Iedere interaktie vergt een hoeveelheid werk van de computer. Hoe meer interakties per tijdseenheid, hoe vaker de computer iets moet doen. Hoe meer beeldschermen, hoe meer interakties. De verwerking op de computer kan per interaktie sterk verschillen en uiteenlopen van het simpel toevoegen van nieuwe order aan een orderbestand tot het bij elkaar zoeken van allerlei gegevens uit diverse bestanden, vergelijkingen, berekeningen enzovoort. Het kriterium voor de belasting die een groep beeldschermen voor een computer betekent is in hoofdzaak het aantal keren dat er iets met een bestand gedaan moet worden: lezen, schrijven of wijzigen. We noemen dat I/O-operaties (Input/Output). Zo kan het uit het oogpunt van de logica van het ontwerp erg verstandig lijken te werken met een groot aantal stuurbestanden, dat zijn bestanden met allerlei min of meer vaste gegevens die door diverse programma's gebruikt kunnen worden. Het lezen en wijzigen van die bestanden kan echter zoveel I/O-operaties vergen, dat daardoor de belasting van het systeem te hoog wordt.
Daarnaast speelt er nog iets dat nog veel ondoorzichtiger is. De computer zoals die geleverd wordt door de computerleverancier werkt ook met allerlei bestanden. Immers, het geheugen in de computer is beperkt en dus moet het beschikbaar geheugen zo efficient mogelijk worden gebruikt. Alles wat niet direkt nodig is voor het operationele gebeuren op een bepaald moment wordt opgeborgen in stuurbestanden. Iedere computerleverancier heeft daarvoor zijn eigen systeem ontworpen. Dat zijn dus ook I/O-operaties, maar nu niet veroorzaakt door de programma's van de eigenaar, maar door die van de leverancier van de computer.
De responsetijd van een interaktie is dus de som van de tijden voor alle systeem-I/O-operaties en voor alle applikatie-I/O-operaties. Hoe meer beeldschermen worden aangesloten, hoe meer interakties het systeem te verwerken krijgt en hoe groter de overhead wordt, uitgedrukt in het aantal systeem-I/O-operaties. Dat betekent onder andere dat ieder computersysteem grenzen heeft, die bepaald worden door het aantal beeldschermen dat wordt aangesloten en de complexiteit van de toepassingen, uitgedrukt in I/O-operaties.
Het aantal beeldschermen is van invloed op de overhead die nodig is om ze te besturen en op het aantal interakties per tijdseenheid. Het zal duidelijk zijn dat het aantal interakties per tijdseenheid afhangt van twee zaken: het aantal interakties per transaktie en de duur van een transaktie. Wanneer er orderentry wordt uitgevoerd op 10 beeldschermen, waarbij iedere transaktie 20 seconden duurt en er 3 interakties met het systeem nodig zijn, dan bedraagt het aantal interakties 3x3x10=90 per minuut. Wanneer

toepassing van de methoden die dit boek behandelt grotendeels moeten zijn verdwenen. Het gaat nu over de tijd die gebruiker moet wachten op een reaktie van de computer. Die tijd noemen we met een slecht nederlands woord: responsetijd. Zeker bij routinewerk ontstaat een bepaald werkritme. Het wordt heel snel als irritant ervaren wanneer dat ritme steeds verstoord wordt. Wanneer de tijd, die gewacht wordt op de reaktie van de computer, niet konstant is, wordt het ritme konstant verstoord. Te lange responsetijden zijn er de oorzaak van dat er helemaal geen ritme in het werken kan ontstaan.
In de praktijk blijken responsetijden van meer dan een minuut voor te komen. Het uiteindelijke tijdverlies kan veel groter zijn dan de som van alle responsetijden. Immers, bij responsetijden van een dergelijke duur gaat die gebruiker andere dingen doen, die vaak langer zullen duren dan de responsetijd. Dat wil zeggen, dat het systeem niet optimaal gebruikt wordt en bijvoorbeeld de dagproduktie niet wordt gehaald (3). Kortom, er zijn allerlei redenen die ervoor pleiten de responsetijden akseptabel te houden.
In sommige gebruikersspecifikaties wordt een poging gedaan, eisen te stellen ten aanzien van responsetijden. Dat gebeurt dan meestal in termen als "de gemiddelde responsetijd moet onder de 2 sec. liggen". Wanneer niets over spreiding wordt gezegd is een gemiddelde responsetijd een nietszeggend begrip. Bovendien: wat is gemiddeld? Gemiddeld over een piekdag? Gemiddeld over een gemiddelde dag? Gemiddeld over alle transakties? Zelfs per transaktie is een gemiddelde responsetijd meestal een nietszeggend gegeven. Sommige responsetijden mogen lang zijn en zullen nooit tot reakties van gebruikers aanleiding geven. Andere responsetijden zijn juist zeer kritisch. Met andere woorden, een gemiddelde responsetijd van een bepaalde toepassing kan 6 seconden bedragen en daarmee buiten de specifikatie vallen. Stel dat het gemiddelde echter ontstaat uit een responsetijd van 2 en een van 10 seconden en dat die 10 seconden voor de gebruiker zeer akseptabel zijn, omdat hij die nodig heeft om bijvoorbeeld dokumenten te paraferen of een dossier af te sluiten. Dan is het geheel voor de gebruiker akseptabel, hoewel de gemiddelde responsetijd er niet aantrekkelijk uitziet. Kortom, wanneer de toepassingen nog ontworpen moeten worden heeft het weinig zin over responsetijden te praten. De eis ten aanzien van responsetijden wordt vaak gesteld in specifikaties ten behoeve van de keuze van computers. In dat stadium is er nog niets ontworpen en hoe kan een computerleverancier nu garanties geven over responsetijden als de transakties nog ontworpen moeten worden?
Laten we de diverse aspekten van responsetijden eens nader bezien. De responsetijd is de tijd die verloopt tussen het indrukken van de laatste, meestal een speciale, toets en het verschij-

hoeft de computer maar eens per 60 seconden zijn werk te doen, evenals het netwerk dat gegevens transporteert. Per transaktie doen computer en netwerk minstens 58 seconden niets voor het beeldscherm waarop de transaktie wordt uitgevoerd. Er kan bepaald worden hoeveel beeldschermen het netwerk en de computer kunnen bedienen. Als bekend is dat een transaktie 1 minuut duurt en de gebruiker kan aangeven hoeveel transakties per uur moeten worden uitgevoerd, is te berekenen hoeveel beeldschermen daarvoor nodig zijn, hoeveel uren gebruikers per dag achter de beeldschermen zitten, wat de throughput van een afdeling is, hoe het zal gaan in pieksituaties, enzovoort. Deze ergonomische resultaten zijn van groot belang voor de gebruiker omdat er tijdens het logisch ontwerp kwantitatief inzicht ontstaat in de uiteindelijke situatie. In de praktijk blijkt steeds weer dat cijfers duidelijker taal spreken dan stapels rapporten of series vergaderingen.
Een ander belangrijk aspekt van Transaktie analyse is, dat het detailschema het rekenmodel van een transaktie is, dat door een rekenprogramma direkt wordt omgezet in resultaten. Met andere woorden, gebruikers hoeven geen cijfers meer te noemen zonder dat ze een idee hebben waar die toe leiden. Als het detailschema er een keer is, kost het praktisch niets om even een aantal situaties door te rekenen. Allerlei gevoeligheidsanalyses zijn nu mogelijk. Zo zijn pieksituaties vaak moeilijk te schatten. Er kan nu zonder meer worden uitgerekend hoeveel uren er extra moeten worden besteed in verschillende pieksituaties.
In de praktijk worden er natuurlijk meer dan een transaktie per terminal uitgevoerd. De informatie-analisten kunnen op basis van de ergonomische resultaten overzichten maken van de bezetting van het benodigde aantal beeldschermen. Transaktie-analisten kunnen op basis van de technische resultaten de belasting van een groep beeldschermen op de computer en het netwerk bepalen.
Samengevat: tijdens het logisch ontwerp ontstaat via dialoogsimulatie voor alle gebruikers een nauwkeurig beeld van alle transakties en via Transaktie analyse worden die beelden kwantitatief in kaart gebracht.

22.6 Responsetijden

Er zijn vele soorten klachten mogelijk over computersystemen. We beperken ons nu tot de toepassingen met beeldschermen, dat zijn interaktieve toepassingen. Immers, de gebruiker typt iets in, de computer reageert, de gebruiker typt iets in, de computer reageert enzovoort. We gaan nu even voorbij aan allerlei irritaties die kunnen ontstaan omdat de gebruiker bijvoorbeeld niet weet wat hij moet intypen of omdat hij een foutboodschap, die op het scherm is verschenen, niet begrijpt. Dat soort problemen zou door

konkrete eisen ontstaan. Het beheer van responsetijden is erg ingewikkeld, maar begint met duidelijke en op de praktijk afgestemde eisen.
De dialoogsimulator kan gebouwd zijn op een microcomputer (22). Als een bedrijf gestandaardiseerd is op een bepaalde microcomputer wordt het simuleren op verschillende vestigingen een eenvoudige zaak: de informatie-analist heeft alleen een paar floppies bij zich. Op iedere werkplek waar een P.C. kan worden opgesteld, kan dialoogsimulatie worden uitgevoerd en daarmee is dan tevens aangegeven dat dialoogsimulatie geen terminalemulatie is. Op het uiteindelijke toetsenbord kunnen de toetsen best wat anders zijn gerangschikt of een andere kleur hebben dan op de P.C. van de simulator. Over dat soort zaken gaan de klachten van gebruikers niet als ze bestaande interaktieve toepassingen evalueren! Aan het eind van de simulatiesessies moet de informatie-analist de transaktieschema's en de schermlay-outs overdragen aan de gebruikers. Voor het management is gemakkelijk na te gaan of de simulatie is uitgevoerd en of de dokumenten zijn overgedragen. Dat is heel wat konkreter dan "het inschakelen van gebruikers".

22.5 Transaktie analyse

Transaktie analyse is een methode om, tijdens het ontwerp, de eisen van gebruikers, die zijn vastgelegd op transaktieschema's, te kwantificeren en om te rekenen naar konsekwenties voor gebruikers, het computersysteem en het netwerk.
De eisen van gebruikers zijn tijdens dialoogsimulatie vastgelegd op transaktieschema's, die door transaktie-analisten worden omgezet in detailschema's. Als er op het transaktieschema staat dat het debiteurennummer moet worden ingetypt, dan wordt op het detailschema aangegeven hoeveel toetsen moeten worden aangeslagen. Als er op het transaktieschema staat dat de gedisplayde gegevens moeten worden vergeleken met wat er op het brondokument staat, wordt op het detailschema het aantal seconden aangegeven dat voor zo'n vergelijking nodig is. Op het transaktieschema geeft een pijl naar rechts aan dat de ingetypte gegevens naar de computer worden gestuurd. Deze interaktie met de computer wordt op het detailschema vastgelegd, tegelijk met het aantal tekens dat verstuurd wordt.
Transaktie analyse levert twee soorten resultaten op: ergonomische en technische. De technische resultaten betreffen de belasting van het netwerk en de computer. De ergonomische resultaten betreffen de terminaltransaktietijd (T.T.T.) en de struktuur ervan. De T.T.T. is de tijdsduur van een transaktie. Als er gedurende een transaktie die 1 minuut duurt, een interaktie met de computer wordt uitgevoerd met een responsetijd van 2 seconden

schreven in gebruikerstaal. Het transaktieschema is tijdens de oplevering dan ook het akseptatiedokument bij uitstek. Is dat misschien de reden dat automatiseerders het zo gemakkelijk vergeten? Wat geldt voor de meeste informatie-analisten, geldt waarschijnlijk ook voor de meeste gebruikers: zet een microcomputer naast een stapel in te vullen transaktieschema's en ze zien alleen de microcomputer.
Op de dialoogsimulator kan slechts de dialoog worden gesimuleerd. Als dat het enige is wat er gebeurt, heeft de informatie-analist na de dialoogsimulatie de schermlay-out en kent hij de dialoog; de gebruiker heeft niets en mag met lege handen de oplevering afwachten. Daarom moet eerst de transaktie worden ontworpen op een transaktieschema. Als dat niet gebeurt, ontaardt dialoogsimulatie gewoonlijk in het vertonen van schermlay-outs en dat kan op elke computer. Bovendien zijn transaktieschema's de startdokumenten voor Transaktie analyse en moeten ze dus toch gemaakt worden.
Dialoogsimulatie biedt de mogelijkheid om het vakmanschap van de gebruikers te koppelen aan het vakmanschap van de automatiseerders. Het is van taktisch belang te kiezen voor de methode dialoogsimulatie. Dialoogsimulatie is een methode omdat de manier van werken na enkele jaren gebruik in de praktijk is uitgekristalliseerd. Er zijn kursussen (15) waarin gebruikers worden voorbereid op het fungeren als mede-ontwerper wanneer de methode dialoogsimulatie wordt toegepast. In kursussen voor informatie-analisten (16) worden de automatiseerders voorbereid op deze vorm van kommunikatie met gebruikers. In beide kurssusen wordt uiteraard ook het verband met de projektaanpak besproken. In de praktijk doet zich steeds weer het probleem voor dat gebruikers zich pas met het ontwerp bemoeien als het systeem gebouwd is of konstant met nieuwe voorstellen komen, zelfs tijdens de bouw. Het ligt voor de hand dat dit laatste elk projekt doet uitlopen. Dialoogsimulatie blijkt dat probleem grotendeels op te lossen: gebruikers hebben nu het systeem ontworpen en ze zien heel duidelijk het afsluiten van het ontwerp en het begin van de bouw.
Een ander aspekt van dialoogsimulatie is de specifikatie van responsetijden. In praktisch alle interaktieve toepassingen zijn de responsetijden het hete hangijzer. Het probleem wordt veroorzaakt door de technische komplexiteit van responsetijden en het feit dat de meest a-technische direkteur van een bedrijf nog kan beoordelen of ze akseptabel zijn of niet. Soms doen gebruikers tijdens het opstellen van de ontwerpspecifikaties pogingen om eisen te stellen aan responsetijden. Meestal noemt men dan gemiddelde responsetijden. Is dat het gemiddelde over alle mogelijke interakties binnen alle transakties? In de praktijk doet geen enkele automatiseerder iets met zo'n getal. Bij dialoogsimulatie kunnen responsetijden worden ingesteld op de simulator, zodat

TRANSAKTIESCHEMA centraal

Menselijke handelingen en bewerkingen	Transport	Machinale verwerking
Begroeten chauffeur Papieren aanpakken en kontroleren: - stempel op uitv.doc - art. 41 dok.aanw. - faktuur aanw. Indien OK: chauffeur wacht. Dossiermap pakken, expeditienr. vaststellen en noteren. Plaatsnemen achter beeldscherm. Op menuscherm kiezen: "INKLARING 31/32"	----)	Selektie eerste scherm INKLARING 31/32. Displayen
Intypen van: - expeditie nr. - zoeknaam	(----	
	----)	Opzoeken volledige N.A.W.

Fig. 22.5 Voorbeeld van een deel van een transaktieschema.

programma's en bestanden. Voor de automatiseerders is prototyping dan ook duidelijk iets anders dan dialoogsimulatie.
Bij de dialoogsimulatie hoeft niet te worden geprogrammeerd en hoeven geen bestanden te worden ontworpen. De simulator beschikt namelijk over een algemeen bruikbare gegevensbank en als er bijvoorbeeld een berekening zou moeten worden uitgevoerd die resulteert in een bedrag, dan wordt een willekeurig bedrag gedisplayed. De dialoog wordt gesimuleerd, niet de berekening.
Alle informatie-analisten en zeker gebruikers van het type D kunnen geheel zelfstandig met een dialoogsimulator werken. Wijzigingen kunnen ter plekke worden aangebracht, alternatieven kunnen dezelfde dan nog getoond worden.
Voor gebruikers als mede-ontwerpers is er weinig verschil tussen dialoogsimulatie en prototyping, voor automatiseerders zijn beide methoden onvergelijkbaar. Omdat prototyping een soort miniprojekt betekent en dus de bekende fasen analyse, ontwerp, bouw en evaluatie inhoudt, kan dialoogsimulatie gebruikt worden voor het ontwerpen van de te prototypen transakties.
De dokumenten die ontstaan tijdens dialoogsimulatie vormen de basis voor Transaktie analyse. De door gebruikers geaksepteerde transakties worden nu vertaald naar kwantitatieve beschrijvingen. Een belangrijk deel van het cijfermateriaal moet geleverd worden door gebruikers. Het rekenprogramma levert resultaten die zowel voor de gebruikers als voor de automatiseerders van belang zijn. De nauwkeurigheid van de resultaten hangt alleen af van de nauwkeurigheid van het cijfermateriaal. Transaktie analyse tijdens het vooronderzoek levert globale cijfers op, tijdens het logisch ontwerp kan de nauwkeurigheid van de ergonomische aspekten binnen 10% liggen. Voor technische aspekten als de responsetijd gaat het om de orde van grootte. In ieder geval wordt duidelijk welke responsetijden langer zullen worden dan 10 seconden.
Tijdens of aan het eind van het technisch ontwerp is voor de responsetijden aan te geven in welke groep ze thuis horen. Deze cijfers dienen door de gebruikers vergeleken te worden met de eisen die aan responsetijden gesteld zijn tijdens dialoogsimulatie.

22.4 Dialoogsimulatie

Dialoogsimulatie maakt deel uit van transaktie-ontwerp. Een transaktie omvat de hele procedure aan het beeldscherm en dat is bijna altijd meer dan intypen en naar het scherm kijken. De dialoogsimulator is een beeldscherm aan een computer of een microcomputer. Als daarmee de schermlay-out en de dialoog zijn bepaald, liggen de overige menselijke handelingen nog niet vast. Daarom begint dialoogsimulatie met het maken van het transaktieschema. Daarop worden de menselijke handelingen en de dialoog be-

Gebruikers van het type C en D specificeren meestal de verwerking en stellen vast welke gegevens daarbij gebruikt moeten worden en wie de eigenaar van de gegevens is, gebruikers van het type A en B ontwerpen de dialoog, in samenwerking met informatie-analisten en eventueel gebruikers van het type C en D.
Een transaktie bestaat uit menselijke handelingen en de dialoog met de computer. Uit een A4 met een schermlay-out is hoogstens de dialoog af te leiden, maar de transaktie is dan nog niet beschreven. De computer bestuurt de dialoog: hij leest wat er wordt ingetypt en zet gegevens op het scherm. Het is duidelijk dat daarom automatiseerders meestal alleen geinteresseerd zijn in de dialoog en de schermlay-out. Voor de gebruiker is het geheel van menselijke handelingen en de dialoog van belang, want dat bepaalt de manier van werken voor de eerstkomende jaren! Gebruikers, die werken met het beeldscherm, ook wel eindgebruikers genoemd, realiseren zich niet kontinu hoe de computer de gegevens verwerkt. Bij transaktie-ontwerp gaat het om de beeldschermkant van interaktieve toepassingen. Bij dialoogsimulatie gaat het alleen om de simulering van de dialoog.Tijdens het logisch ontwerp kunnen met een dialoogsimulator de toekomstige transakties volledig worden gesimuleerd en beoordeeld. Na dialoogsimulatie beschikken automatiseerders over geaksepteerde dialoogspecifikaties en schermlay-outs.
Een andere term die in dit verband vaak genoemd wordt is prototyping. Bij prototyping bouwt men met bepaalde hulpmiddelen in korte tijd een werkend systeem. Daarin zijn natuurlijk niet alle details opgenomen, want dan zou het hele uiteindelijke systeem in die tijd gereed zijn en zou prototyping overbodig zijn. Er wordt een deel van het systeem gebouwd, de kern, de belangrijkste funkties of van alle funkties de grote lijnen, er is dus een computer nodig met beeldschermen. Als daarop het prototype gereed is, dan zijn er dialogen, programma`s en bestanden ontworpen en gebouwd. Als de beeldschermen verplaatsbaar zijn kan op iedere werkplek geprototyped worden, zo niet, dan gaan de gebruikers naar de beeldschermen. In principe kan er nu hetzelfde gebeuren als bij dialoogsimulatie. De volledige transaktie kan gesimuleerd worden, de computer zal zelfs een deel van de berekeningen uitvoeren en de juiste gegevens op het scherm zetten. Wanneer de gebruikers met andere voorstellen komen of alternatieven willen beoordelen, moeten de programma`s en de schermlay-outs worden aangepast. Daar is natuurlijk enige tijd mee gemoeid. In principe gaat alles net als bij iteraties over de life cycle van het projekt heen alleen het gaat sneller omdat er maar een subset van alle programma`s aanwezig is en er bovendien middelen zijn om die programma`s snel te genereren. Na prototyping beschikken de automatiseerders over geaksepteerde dialoogspecifikaties en schermlay-outs, geteste

staan op ponskaarten, tapes, floppies, schijven en dergelijke. Een uitvoerbestand kan staan op papier, tapes, floppies, schijven en dergelijke. Zo worden bijvoorbeeld eerst alle orders ingelezen en op een schijf geschreven, vervolgens worden alle orders op de schijf vergeleken met het debiteurenbestand en daarna verwerkt door een programma dat het voorraadbestand aanpast. Zo worden steeds grote hoeveelheden, batches, van dezelfde soort gegevens verwerkt in stappen. Het resultaat van de ene stap is invoer voor de volgende stap.
De verwerking bestaat uit twee delen: het lezen en schrijven van gegevens en het uitvoeren van rekenwerk volgens een bepaald algoritme. In principe moet de gebruiker die twee zaken bepalen. Hij moet aangeven welke algoritmes moeten worden uitgevoerd en met welke gegevens. Als de computer de minimale voorraad moet berekenen, moet bepaald worden hoe er gerekend moet worden en met welke gegevens.
Veel systeemontwikkelingsmethoden die eigenlijk bedoeld zijn voor batch-verwerking, worden vaak ook toegepast voor interaktieve toepassingen. Voor automatiseerders is het verschil tussen batch-verwerking en dialoogverwerking of transaktieverwerking minimaal. In- en uitvoer worden gekombineerd tot dialoog en de lay-outs van de lijsten worden nu de schermlay-outs. De verwerking is iets anders geworden, omdat nu per transaktie alle stappen doorlopen worden. De gebruiker wordt nu dus ingeschakeld om de schermlay-outs te beoordelen. In de kommunikatie met de gebruikers is niet veel veranderd. Er is zo langzamerhand voldoende geschreven over problemen die gebruikers hebben als ze met beeldschermen moeten werken. Wij zullen dat niet herhalen. Er bestaat een hemelsbreed verschil tussen het gebruiken van lijsten die door een computer zijn vervaardigd en het interaktief bedienen van een computer. Daarbij gaat het niet zozeer om de angst voor het toetsenbord: met de invoering van microcomputers op scholen en het groeiende aantal P.C.'s in de huiskamer, gaat dat snel over. Het gaat gewoon om de manier waarop een beeldscherm deel uitmaakt van het dagelijkse werk. De angst voor een toetsenbord, zo die al bestaat, wordt in de praktijk vaak snel overwonnen door de ergenis dat bepaalde zaken niet kunnen of veel te onhandig zijn.
Veel automatiseerders denken nog te vaak dat zij het beste weten wat goed is voor gebruikers en bijna alle gebruikers denken dat het ontwerpen van informatieverwerkende systemen een zaak is van de automatiseerders. Het ontwerpen van transakties is echter voor minstens 80% een zaak van gebruikers. Wie moet er straks werken met het beeldscherm? Natuurlijk moet er een keer worden vastgesteld welke berekeningen de computer moet uitvoeren en welke gegevens daarbij nodig zijn. De dialoog is een zaak van de gebruikers die later een beeldschem zien verschijnen op hun werkplek.

Hoe een bedrijf ook is georganiseerd, bij voldoende detaillering komen we terecht bij funktionarissen. In dit verband kan dat zowel een direkteur als een medewerker zijn, in ieder geval een persoon, met een taakstelling, plannen voor de toekomst, een mening over het niveau van zijn werk, enzovoort.
Wat er in een bedrijf ook gebeurt, bij voldoende detaillering komen we terecht bij procedures zoals die door een funktionaris worden uitgevoerd. De procedures worden geheel of gedeeltelijk vervangen door transakties. Aan het eind van het logisch ontwerp weet iedere funktionaris precies hoe zijn werkplek er uit zal gaan zien, welke procedures en welke transakties nu tot zijn werk behoren. Kortom, tijdens het logisch ontwerp kan de nieuwe taak/funktie-omschrijving al worden gemaakt.

22.3 Interaktieve toepassingen

In de periode voor de invoering van beeldschermen verwerkten computers invoerbestanden. Pakken ponskaarten werden gelezen en verwerkt, resultaten werden direkt of op kommando afgedrukt op lijsten. Voor de gebruiker was de computer een apparaat dat gegevens las en gegevens afdrukte. Met de invoer had hij meestal weinig te maken omdat bestaande dokumenten werden verponst op de ponskamer. De afgedrukte resultaten waren voor hem bestemd, dus had hij inspraak in de manier waarop de gegevens werden afgedrukt. Informatie-analisten brachten de bestaande situatie in kaart, bespraken met de gebruikers de lay-out van de lijsten en gingen aan het werk.
Doordat het werk in principe vaak op hetzelfde neer kwam en er toch steeds dezelfde fouten werden gemaakt, ontstonden systeemontwikkelingsmethoden: daarin is nauwkeurig beschreven hoe de diverse automatiseerders dienen te werken. In die methoden kwam altijd dit punt voor: bepalen van de in- en de uitvoer. Dat betekende dus enig overleg met gebruikers.
Een ander karakteristiek aspekt van die periode was het feit dat het vaak ging om de automatisering van administratieve aktiviteiten, zaken waarvan soms de informatie-analisten meer verstand hadden dan sommige administratieve medewerkers. Met andere woorden, de informatie-analisten hadden vaak maar een half woord nodig om te begrijpen waar het de gebruikers om ging.
De meeste systeemontwikkelingsmethoden zijn dan ook nog gebaseerd op deze manier van verwerking, bekend als batch-verwerking.
Er wordt nog steeds veel batch-werk gedaan op computers. Een computer die is uitgerust met beeldschermen voert vaak batch-werk uit als er niets te doen is voor de beeldschermen.
Karakteristiek voor batch-verwerking is dus: invoer van een bestand, verwerking, uitvoer van een bestand. Een invoerbestand kan

natuurlijk alleen als een methode als transaktie-ontwerp wordt ingevoerd. Dan vindt de akseptatietest door de gebruikers plaats tijdens het logisch ontwerp. In de praktijk hebben we meestal te maken met een aantal projekten voor een systeem. In Fig. 22.3 is de aanpak nog eens kort weergegeven voor verscheidene projekten. Per projekt leidt transaktie-ontwerp tot de akseptatietest tijdens het logisch ontwerp. Projekten worden tot en met het logisch ontwerp in opdracht gegeven en wanneer van de groep de logische ontwerpen gereed zijn, zijn de oplossingen geaksepteerd door de gebruikers. Dan is het ontwerp met voldoende nauwkeurigheid vastgelegd om een goede kostenkalkulatie uit te voeren.
De verschuiving zoals die in de figuur is aangegeven, is natuurlijk niet noodzakelijk. Als er voldoende P.C.'s en informatieanalisten beschikbaar zijn kunnen de logische ontwerpen parallel worden uitgevoerd.
Tijdens het logisch ontwerp kan het laatste woord over de performance nog niet gezegd worden. Aan het eind van het technisch ontwerp moeten de ontwerpers per transaktie definitief aangeven of ze aan de gestelde eisen kunnen voldoen.
Als de automatisering wordt aangepakt volgens de beschreven methoden komen tijdens het logisch ontwerp ook de relaties met andere bedrijfsaspekten naar voren, zoals is aangegeven in Fig. 22.4.

BEDRIJF			
Organisatie	Funkties	Geografie	Automatisering
↓	↓	↓	↓
Funktionaris	Procedures	Werkplek	Transakties

Fig. 22.4 Transaktie-ontwerp in het grote geheel.

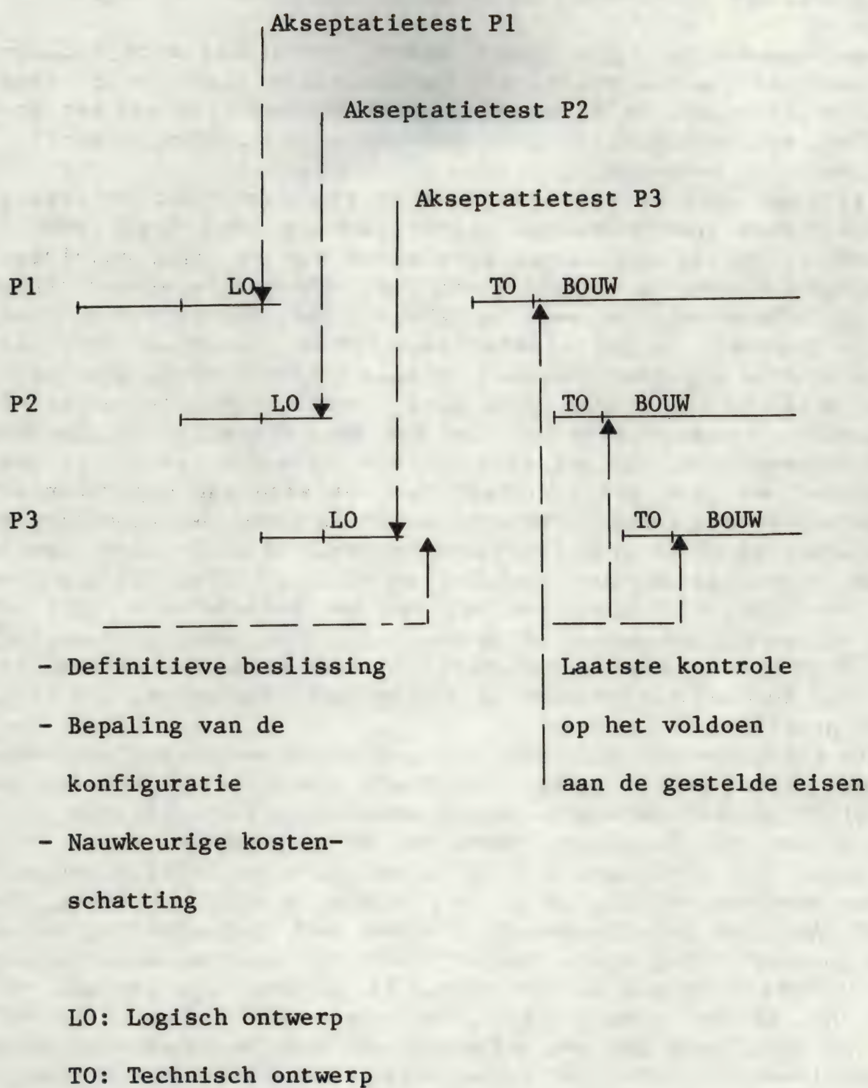


Fig. 22.3 Akseptatietest tijdens het logisch ontwerp

ren gebruikers de toekomstige situatie al tijdens het logisch ontwerp.
In systeemontwikkelingsmethoden waarin gebruikers worden gekonfronteerd met automatiseringsdokumenten en de rest van de transaktie mogen raden, is de akseptatietest aan het eind van het projekt het scharnierpunt. Zouden daardoor zoveel projekten nooit aan invoering toekomen?
Als tijdens het logisch ontwerp alle gebruikers met het beeldscherm hebben kunnen werken, blijkt vaak dat veel angst voor de automatisering is weggenomen en daarmee ook de angst om cijfers te noemen. Waneer de invoering van beeldschermen negatieve effekten zal hebben voor het werk of de werksfeer, dan zijn die aspekten nu konkreet en nog steeds bespreekbaar, omdat er voor dialoogsimulatie nog geen computer in huis hoeft te zijn. Dialoogsimulatie kan op elk moment en op elke lokatie worden gerealiseerd. De sociale aspekten zijn nu voor iedereen duidelijk en dus ook goed bespreekbaar. Aan het eind van het logisch ontwerp ligt precies vast wat gebruikers willen. Aan het eind van het technisch ontwerp ligt een tweede scharnierpunt: dan mogen de automatiseerders aangeven of de gestelde eisen haalbaar zijn. Zo niet, dan is er nog niets aan de hand, wanneer gebruikers afzien van een aantal transakties of misschien wel van het hele projekt. Als het gaat om een nieuwbouwprojekt op een nog aan te schaffen computer, dan is er wat tijd en papier verloren maar er is nog een weg terug. Als de computer eenmaal is aangeschaft bestaat die weg terug in de praktijk niet meer.
Iedere gebruiker zou deze scharnierpunten in een automatiseringsprojekt moeten kennen en hij zou steeds moeten weten in welke fase "zijn" projekt verkeert. Dat is belangrijker dan alle technische kennis over computers, bestanden en programma`s.
Het maken van iets ingewikkelds is meestal een iteratief proces. Na de probleemstelling volgt een eerste ontwerp dat wordt gebouwd. Allerlei tekortkomingen zijn dan vaak de aanleiding om nog eens opnieuw te beginnen. Ook het invoeren van automatisering zou een iteratief proces moeten zijn. In de praktijk gebeurt dat niet. Het eerste ontwerp blijft het definitieve produkt. Een blik op Fig. 22.2 geeft aan hoe dat komt: per fase is aangegeven welke automatiseerders erbij betrokken zijn. Aan het eind van iedere fase gaan de betrokkenen over naar een ander projekt en bij ingehuurd personeel gaan ze zelfs de deur uit. Met wie evalueren de gebruikers nu het resultaat van een projekt? Voor veel gebruikers zijn overdracht en evaluatie het begin van het nieuwe leven. De automatiseerders zijn dan niet meer geinteresseerd, die zijn allang met een volgend projekt bezig. Dit beeld geeft nog eens aan waarom het ontwerpproces iteratief moet zijn: dan zijn dezelfde automatiseerders betrokken bij ontwerp en evaluatie. Dat werkt

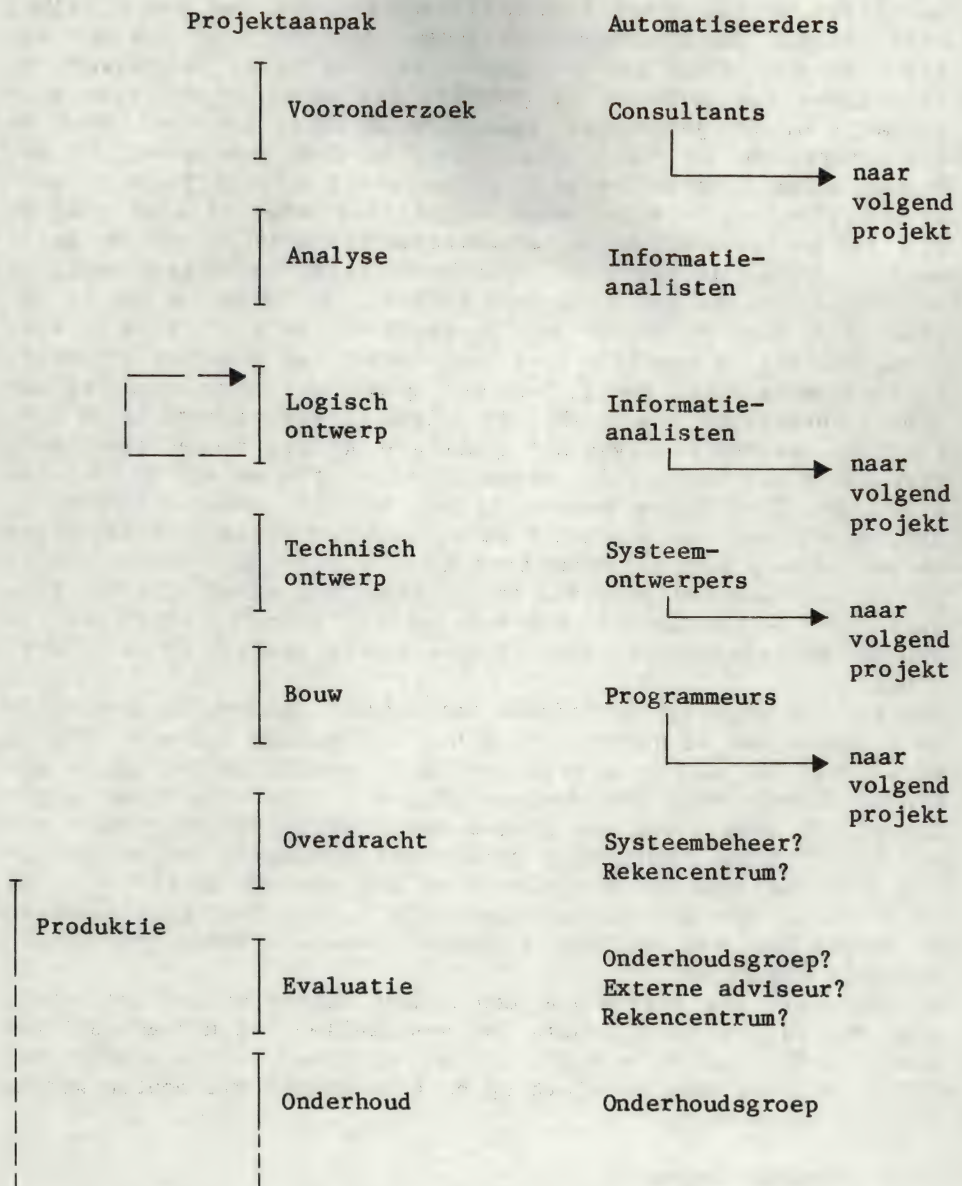


Fig. 22.2 Het iteratieve proces

gebruiker worden ingeschakeld. Niet op een vrijblijvende manier, zoals een willekeurige informatie-analist dat het beste lijkt, maar volgens overeengekomen methoden. Aan het eind van het logisch ontwerp dient het management zich ervan te vergewissen of er volgens die methoden is gewerkt. Dan wordt er dus niet meer gevraagd of de gebruikers ingeschakeld zijn, maar wel naar de deelnemers aan de dialoogsimulatie, de data waarop dat is gebeurd, welke transakties zijn gesimuleerd, welke cijfers er uitgerold zijn, welke konklusies daaruit getrokken zijn en waar de bij de methode behorende gebruikersdokumenten zijn. Op basis daarvan worden de gestelde eisen beoordeeld. Vervolgens beginnen de ontwerpers aan het technisch ontwerp. In feite is dat de poging om althans op papier aan de eisen van de gebruikers te voldoen. Daarbij is geweldig veel vakmanschap van allerlei automatiseringsspecialisten nodig, voor de gebruiker is het meestal een geheel onbegrepen proces. Hij is alleen geinteresseerd in de konklusies. Aan het eind van het technisch ontwerp moeten de automatiseerders per transaktie aangeven in hoeverre ze aan de gestelde eisen wat dialoog en responsetijden betreft kunnen voldoen. Op dat moment heeft de gebruiker de vrijheid om bepaalde transakties te annuleren, naar alternatieven te zoeken of zijn eisen aan te passen aan de mogelijkheden. Na het technisch ontwerp is het projekt voor de gebruikers, wat hun inbreng betreft, afgelopen. Ze hebben hun toepassingen besteld en mogen nu wachten op de aflevering.
Doordat het mogelijk is tijdens het logisch ontwerp de toekomstige situatie wat de procedure aan het beeldscherm betreft volledig na te bootsen, kan er alvast een begin gemaakt worden met de gebruikershandleiding, de voorbereiding van organisatorische wijzigingen, de opstelling van nieuwe taak/funktie-omschrijvingen enzovoort. Tijdens de oplevering nemen de gebruikers hun dokumentatie weer ter hand en kontroleren of het systeem werkt zoals het is besteld. Het akseptatiedokument is in hun taal geschreven en ze zouden het zelf geschreven kunnen hebben. Zo wordt een echte kontrole mogelijk.
De evaluatie zal altijd nog een aantal nieuwe wensen opleveren, maar dat is heel iets anders dan vaststellen dat niemand dit besteld heeft, dat de gebruikersvriendelijkheid ver te zoeken is, dat de dialoog niet aansluit op de handelingen die moeten worden verricht enzovoort.

22.2 De scharnierpunten

Het scharnierpunt is letterlijk het punt waar het om draait. Voor gebruikers draait alles om het logisch ontwerp. Als methoden als dialoogsimulatie en Transaktie analyse worden toegepast, evalue-

gaat het uiteraard niet alleen om de interaktieve toepassingen, maar om alle funkties. Welke waarde kan worden gehecht aan een kosten/baten-analyse hangt van een groot aantal meestal vage faktoren af. In de praktijk rekent men vaak met het dubbele van de geschatte kosten. Als echter het aksent ligt op de interaktieve toepassingen en het te ontwerpen netwerk, is het aan te bevelen om het transaktie-ontwerp voor de belangrijkste transakties alvast uit te voeren tijdens het vooronderzoek. Dit kan bijvoorbeeld het geval zijn wanneer het projekt staat of valt met de kosten van het netwerk. Schattingen van dat soort kosten zijn meestal nergens anders op gebaseerd dan op zogenaamde ervaring. Ervaring op het gebied van nog te ontwerpen interaktieve toepassingen is echter dun gezaaid. Als die ervaring zou bestaan, zou toch in bestaande situaties bekend moeten zijn hoe die lange responsetijden nu eigenlijk ontstaan. Bij evaluatie van bestaande systemen blijkt steeds weer dat niemand weet waar het knelpunt zit. Schattingen ten aanzien van kosten van nog te ontwerpen netwerken kunnen we dus beter vergeten. Als die cijfers bekend moeten zijn tijdens het vooronderzoek dan moet het transaktie-ontwerp worden uitgevoerd tijdens het vooronderzoek. Wat dat voor de automatiseerders betekent wordt behandeld in het deel voor de informatie-analisten. Voor de gebruikers maakt het niet veel uit wanneer het transsktie-ontwerp wordt uitgevoerd.

In het algemeen is het vooronderzoek om allerlei redenen echter niet het aangewezen moment om te gaan ontwerpen. Zolang er maar niet teveel waarde wordt gehecht aan de zogenaamde kosten/baten-analyse is een vooronderzoek best zinvol. Het vooronderzoek wordt meestal afgesloten met een GO/NO GO beslissing. Zolang die beslissing maar niet teveel gebaseerd is op de kosten/baten-analyse kan het geen kwaad. In het vervolg zal blijken dat er betere momenten zijn voor dergelijke beslissingen.

Dan volgt de fase waarin de processen en gegevens in kaart worden gebracht: de bestaande situatie wordt geanalyseerd. Als gebruikers zich bewust zijn van de fasen die nog volgen hoeven ze zich niet al te veel zorgen te maken over de cijfers die ze verstrekken.

Wanneer de bestaande aktiviteiten en procedures aldus zijn vastgesteld, begint langzamerhand het ontwerpproces. Iedere informatie-analist met enige ervaring heeft tijdens de analyse natuurlijk af en toe al gedacht aan de geautomatiseerde versie van handmatige procedures. Sommige procedures zullen helemaal vervallen omdat ze door een programma in de computer zullen worden uitgevoerd, andere procedures zullen vertaald worden in transakties en weer andere procedures zullen uitgevoerd worden aan de hand van lijsten, die door de computer worden geproduceerd.

In alle situaties waarin aan beeldschermen wordt gedacht moet de

Automatiseerders	Gebruikers
Vooronderzoek	Globale kosten/baten-analyse
Analyse	Bestaande situatie in kaart brengen
Logisch ontwerp	Nieuwe situatie ontwerpen
Technisch ontwerp	Kontrole of aan de eisen is voldaan
Bouw	Voorbereiding invoering
Oplevering	Kontrole aan de hand van gebruikersdokumenten
Evaluatie	Evaluatie

Fig. 22.1 Fasen binnen de projektaanpak.

Hoofdstuk 22
Projektaanpak

22.1 Fasen

In de meeste bedrijven wordt een automatiseringsprojekt in een aantal fasen gerealiseerd. De opeenvolging van die vastgestelde fasen wordt de projektaanpak genoemd. Iedere fase is op zich meestal weer zo ingewikkeld dat er behoefte bestaat om per fase de aktiviteiten en de te produceren dokumentatie te rubriceren. Per aktiviteit kunnen methoden en middelen worden bepaald. Soms is de gehele projektaanpak zo uitgekristalliseerd, gestandaardiseerd en gedokumenteerd dat er van een systeemontwikkelingsmethode gesproken wordt. Hoewel het gaat om een groot aantal aktiviteiten waar de gebruiker niets mee te maken heeft, is de indeling in fasen toch van groot belang voor de kommunikatie met de automatiseerders. Als de projektaanpak op geen enkele manier is vastgelegd, kunnen gebruikers wel vergeten, dat ze ooit vat zullen krijgen op de automatisering. De projektaanpak is voor de gebruikers om twee redenen van belang. Ze behoren vast te kunnen stellen wanneer er inbreng van hen wordt verwacht en ze moeten de momenten kennen waarop gekontroleerd kan worden of er aan hun eisen voldaan zal worden.
In de meeste systeemontwikkelingsmethoden zijn de fasen zoals is aangegeven in Fig. 22.1. Tijdens het vooronderzoek wordt van de gebruikers verwacht dat ze in grote lijnen aangeven welke funkties er verwacht worden van het computersysteem. Op dat moment

gereed zijn. Gebruikers verergeren zelf vaak dit soort situaties door meteen ook nog een exakte prijs te willen hebben voor het ontwerp, de bouw, de hardware en de exploitatiekosten!
Bij het opzetten van automatisering gaat het wat de gebruikers betreft om twee belangrijke fasen:

- analyse van de bestaande situatie
- logisch ontwerp van de geautomatiseerde situatie.

Wanneer gebruikers alleen betrokken zijn bij de analysefase, dan wordt in principe de bestaande situatie geautomatiseerd. Wanneer de gebruiker ook op de juiste wijze wordt ingeschakeld bij het logisch ontwerp, kan er een belangrijke toegevoegde waarde aan de automatisering ontstaan. Immers, wie zou beter nieuwe mogelijkheden voor de manier van werken kunnen bedenken dan de gebruiker zelf? Zeker naarmate het minder gaat om de standaard administratieve toepassingen, neemt de deskundigheid en dus de inbreng van informatie-analisten snel af.
Samengevat kan worden vastgesteld dat het taktisch gebruikersmanagement ervoor moet zorgen, dat de bestaande situatie goed in kaart kan worden gebracht en dat gebruikers volgens effektieve ontwerpmethoden worden ingeschakeld bij het logisch ontwerp. Over dat laatste gaat het in de volgende hoofdstukken.

katies in vast te leggen maar die kan de gemiddelde gebruiker niet lezen. Maar dan nog is er kontinu ruggespraak nodig over wat er nu werkelijk bedoeld wordt.
Het gaat eigenlijk om drie zaken:
- Hoe stellen we als gebruikers vast wat we eigenlijk willen?
- Hoe leggen we dat vast voor gebruikers en automatiseerders?
- Hoe zorgen we ervoor dat dat gebouwd wordt?

Voor geen van de drie aspekten is automatiseringskennis nodig op het niveau van automatiseerders. De betrokkenheid van het taktisch gebruikersmanagement hoeft, wat de specifikaties voor interaktieve toepassingen betreft, dus slechts het vast te stellen of de methoden die worden toegepast voor bovengenoemde drie terreinen, effektief kunnen zijn in de betreffende gebruikerssituatie. Om dat te kunnen beoordelen zijn presentaties, demonstraties of proefprojekten noodzakelijk. Voor de betrokken gebruikers moet vaststaan dat het werkt.
Er is nog een andere kant aan de zaak en dat is de gebruikerswereld zelf. Er zijn vele bedrijven die uitstekend draaien, maar waarvan eigenlijk alleen het administratieve proces goed is vastgelegd in een handboek. Allerlei operationele processen in het bedrijf liggen alleen vast in de hoofden van de ervaren medewerkers. Met name in allerlei dienstverlenende bedrijven is dat het geval. Dat betekent, dat voor er aan automatisering wordt gedacht, eerst de eigen processen en gegevens in kaart moeten worden gebracht. Dat wil zeggen: kwalitatief, om welke processen gaat het, en kwantitatief, om welke frekwenties en welke aantallen per dag gaat het. Wanneer de eigen situatie niet goed in kaart is gebracht, is een vertaalslag naar een computersysteem met beeldschermen niet mogelijk. Dan ontstaat zeer zeker de situatie waarin een slimme, interne of externe informatie-analist twee dagen in het bedrijf rondloopt, het aantal beeldschermen bepaalt aan de hand van het aantal medewerkers, het aantal schijven aan de hand van het aantal postbakjes en de geheugengrootte aan de hand van de leeftijd van de afdelingschefs. Overdreven, ja, maar het lijkt er vaak veel op, omdat er in twee dagen geen behoorlijke analyse van de bestaande situatie kan worden gemaakt. Met name in het midden- en kleinbedrijf denkt het management vaak over automatisering in termen van aanschaf van een computer en de bouw of koop van een hoeveelheid programma's. Analyse van de eigen situatie in een vorm, die kan dienen als basis voor de automatisering is volledig onbekend. Er wordt vaak bezuinigd op analyse en ontwerp, omdat dat alleen maar geld kost. Wat koop je voor al die ordners met dokumentatie? Programma's zijn ook wel duur maar die laten de computer tenminste werken. Dat is nu typisch de verkeerde manier van zuinig zijn en de oorzaak van de altijd tegenvallende kosten, omdat de programma's jarenlang bijna

voor 95% uit het voeren van de dialoog, andere voor slechts 10%. Als een aktiviteit van een gebruiker bestaat uit het uitvoeren van een aantal procedures, dan zal in de geautomatiseerde situatie een deel van de procedures zijn vervangen door een of meer transakties. Bijna altijd moeten transakties goed aansluiten bij de resterende of nieuwe procedures. Bij het betrekken van de gebruikers in het ontwerpproces gaat het om de kwalitatieve en kwantitatieve aspekten van het transaktie-ontwerp.
Binnen het transaktie-ontwerp dient de methode dialoogsimulatie voor het scheppen van de uiteindelijke situatie tijdens het ontwerpproces en de methode Transaktie analyse brengt de ontworpen transakties kwantitatief in kaart. Transaktie analyse levert ergonomische gegevens, die voor gebruikers van belang zijn en daarnaast technische gegevens, aan de hand waarvan de automatiseerders uitspraken kunnen doen over responsetijden, kosten van een netwerk en de grootte van de konfiguratie. Indirekt zijn dus ook technische resultaten toch weer van belang voor de gebruikers. Fig. 21.3 brengt een en ander in beeld. Of deze methode de gewenste resultaten opleveren is voor iedereen kontroleerbaar, omdat er al jaren kursussen worden gegevens (15, 16 en 17), waarin ze worden behandeld.

21.5 Eigen betrokkenheid

In talloze bedrijven voelen gebruikers zich ongelukkig met de komende of zich uitbreidende automatisering. Bovendien is automatisering een vak apart, dat door specialisten wordt uitgeoefend. De resultaten van automatisering zijn echter zeer definitief en beinvloeden in hoge mate de werksfeer. Het is maar de vraag of het werken met een beeldscherm nuttig, leuk of interessant is. Gebruikers staan dan ook meestal niet te trappelen om mee te werken aan de opbouw van de automatisering. Het resultaat is dat automatiseerders doorgaan met wat ze al dertig jaar gewend zijn: volkomen zelfstandig systemen ontwerpen voor gebruikers. De vraag waar het taktisch gebruikersmanagement voor staat is eigenlijk deze: Hoe laat ik gebruikers specificeren wat ze willen, zonder dat ze "zelf automatiseerders worden"? Nog korter gezegd: "Hoe laat ik gebruikers de gebruikersspecifikaties opstellen"? Iedereen die enige ervaring heeft in het op schrift stellen van de specifikaties van bijvoorbeeld een beeldschermtransaktie kent de problemen van de gesprekken met lekengebruikers die zich bij het woord beeldscherm geen konkrete handelingen voor kunnen stellen. Daarnaast is het praktisch onmogelijk in al dan niet gespierd proza specifikaties te maken, die kompleet zijn en niet voor verschillende uitleggingen vatbaar. In de praktijk lijden dit soort pogingen schipbreuk. Er zijn speciale talen ontwikkeld om specifi-

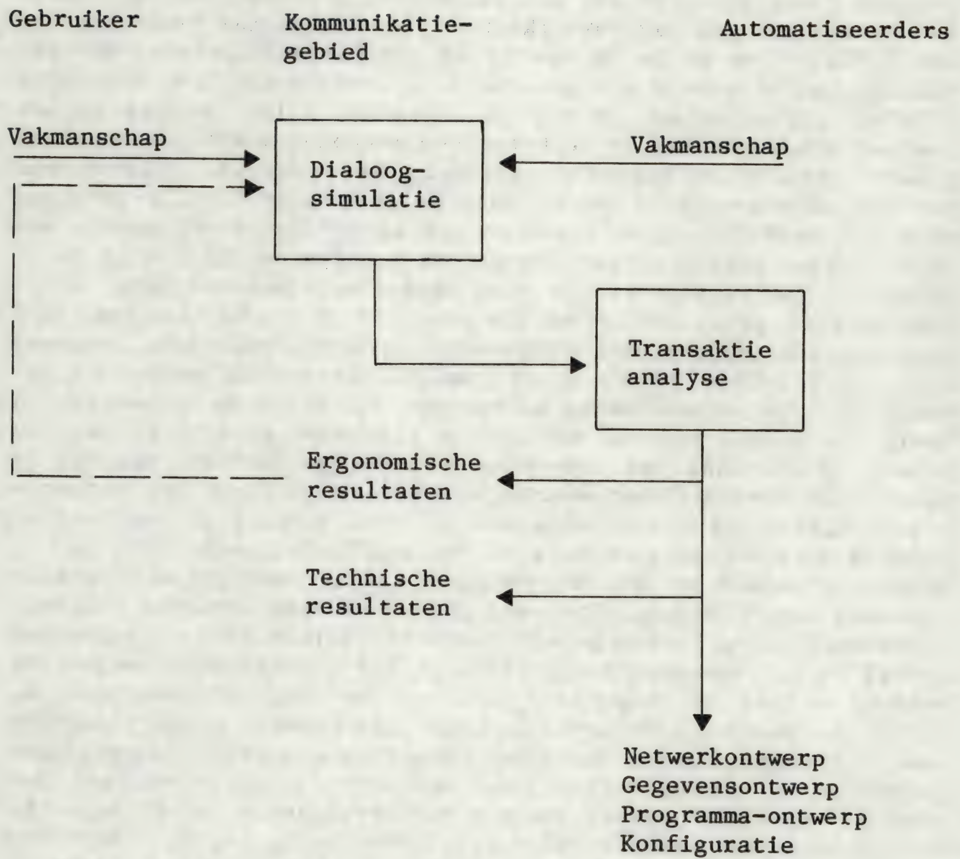


Fig. 21.3 Transaktie-ontwerp

ogen eenvoudige aanpassingen. De methoden moeten daarin verbetering brengen. Het moet mogelijk worden de toekomstige situatie nu te evaluaren!
Een ander probleem voor gebruikers is het verstrekken van gegevens over hun werk, zonder te weten tot welke gevolgen dat leidt. Het zal duidelijk zijn dat er een verband bestaat tussen het aantal orders dat per dag verwerkt wordt en het aantal beeldschermen dat nodig is om ze in te voeren in een computersysteem. Een gemiddeld aantal orders is vaak moeilijk vast te stellen, bovendien zijn er pieksituaties en wil de gebruiker enige reserve in het systeem inbouwen. Met de te bespreken methoden wordt het mogelijk eisen, cijfers en gegevens van gebruikers direkt, volgens een vaststaand rekenproces om te rekenen naar bijvoorbeeld gevolgen voor het aantal beeldschermen en het aantal beeldschermuren. Andere cijfers leiden direkt tot nieuwe konklusies. Zo krijgt de gebruiker gevoel voor kritisch en minder kritische cijfers.
Een volgend probleem is de dokumentatie die moet fungeren als kommunikatiemiddel tussen gebruikers en automatiseerders. Automatiseerders hebben tot taak de gebruikerssituatie geheel of gedeeltelijk te automatiseren en brengen dus alles op hun manier in kaart. De meeste van die dokumenten zijn voor gebruikers niet te lezen. Zo ontstaat een het-zal-wel-goed-zijn-houding, waarmee de gebruikers zichzelf het meeste kwaad doen. Tijdens het ontwerpproces moeten dokumenten ontstaan die voor gebruikers leesbaar zijn en waar automatiseerders verder mee kunnen werken.
Tenslotte noemen we nog het probleem dat er voor gebruikers vaak geen weg meer terug is. Stel dat gebruikers een bepaalde interaktieve toepassing hebben gedefinieerd die alleen goed te gebruiken is bij korte responsetijden. In de praktijk wordt zo`n toepassing gebouwd en als de responsetijden veel te lang blijken te zijn, zullen de automatiseerders hoogstens, in meestal onbegrijpelijke taal, uitleggen waarom het niet anders kon. Hoeveel toepassingen zouden er al gerealiseerd zijn, die nooit gebruikt worden? Het beheer van de performance van een computersysteem is een ingewikkeld proces. Wat daar in ieder geval voor nodig is, zijn methoden om gebruikers konkrete eisen te laten stellen en om die eisen te vertalen naar gevolgen voor gebruikers en voor het computersysteem. Daarom moeten de te behandelen methoden zowel door gebruikers als door automatiseerders beoordeeld worden op hun resultaten.
In het kader van het ontwerpen van interaktieve toepassingen gaat het om methoden die leiden tot het transaktie-ontwerp. Een transaktie is een beeldschermvesie van een handmatige procedure. De handmatige procedure bestaat uit menselijke handelingen, een transaktie bestaat uit menselijke behandelingen en een dialoog met de computer via het beeldscherm. Sommige transakties bestaan

ontwerpproces de te bouwen situatie zien en ervaren en waarmee de gevolgen van eisen voorgerekend worden.

21.4 Waarom methoden?

Een methode is een uitgekristalliseerde, gestandaardiseerde en goed gedokumenteerde manier van werken. Uitgekristalliseerd betekent in dit verband dat er ervaring is opgedaan met deze manier van werken. Het wiel hoeft niet opnieuw te worden uitgevonden en alle beginnersfouten worden voorkomen. Standaardisatie betekent dat de manier van werken voor iedereen geldt. De standaard hoeft niet star te zijn, binnen beschreven grenzen kunnen best enkele vrijheidsgraden bestaan. De manier van werken is goed gedokumenteerd en daarmee overdraagbaar. Veel automatiseerders hebben een hekel aan methoden want ze voelen zich erdoor beperkt in hun kreatieve mogelijkheden. Als informatie-analisten zelf mogen bepalen op welke wijze ze gebruikers inschakelen bij het ontwerpproces dan zal de samenwerking per analist per projekt anders zijn. De vraag of gebruikers zijn ingeschakeld, kan ten allen tijde met "ja" beantwoord worden, zonder dat vaststaat met welk resultaat. Het is in het belang van gebruikers dat de kommunikatie tussen automatiseerders en gebruikers volgens een gekozen methode verloopt en het is de enige manier om de kommunikatie te managen. Omdat het gaat om de koppeling van twee geheel verschillende vakgebieden moet de keuze voor methoden gedragen worden door de gebruikers en de automatiseerders. Gebruikers moeten beoordelen of de methoden het mogelijk maken vat te krijgen op het ontwerpproces, automatiseerders beoordelen de methoden hoofdzakelijk op de aansluiting op andere ontwerpmethoden die ze al gebruiken. We zullen nu alleen de gebruikersaspekten verder uitwerken.

Het grootste probleem voor gebruikers is het feit dat ze moeten kiezen voor een systeem dat ze nog niet kennen. Gebruikers van het type C en D kunnen meestal heel goed de funktie van het systeem beschrijven. Voor het ontwerpen van de procedure aan het beeldscherm is de inbreng van gebruikers van het type A en B nodig. In de praktijk wordt de gebruikers een map met schermlayouts getoond en wordt van hen verwacht dat ze zich akkoord verklaren met wat er ontworpen is. Problemen doen zich dan ook meestal voor tijdens de invoering, omdat de gebruikers dan pas echt met het systeem gaan werken. Volledigheidshalve is in de meeste systeemontwikkelingsmethoden nog de fase evaluatie opgenomen waarin de gebruikers hun kommentaar kunnen geven op het in produktie zijnde systeem. Of dat kommentaar ooit nog een keer wordt vertaald naar een aanpassing van het systeem, is de vraag. In vele bedrijven wachten gebruikers al jaren op enkele, in hun

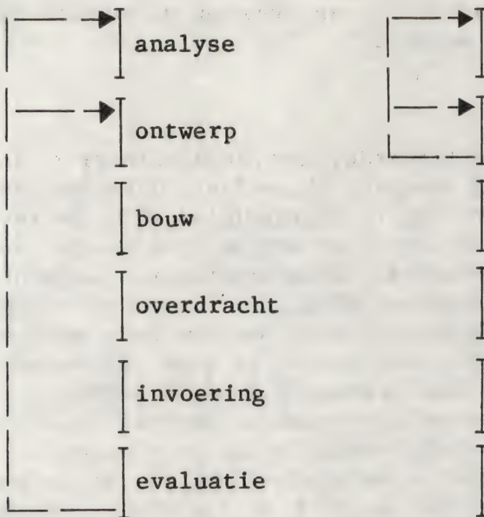


Fig. 21.2 Het iteratieve ontwerpproces.

gen worden gedaan die worden omgerekend naar konsekwenties. Als de konsekwenties niet akseptabel zijn, moeten de eisen worden aangepast. Met de nieuwe eisen worden de nieuwe konsekwenties vastgesteld. Dit proces wordt herhaald totdat beide akseptabel zijn. In de praktijk komt er van dat itereren niet veel terecht. Gebruikers stellen eisen en automatiseerders bouwen een systeem gebaseerd op die eisen en pas na de oplevering blijkt of het systeem voldoet aan de eisen. Als blijkt dat gebruikers iets anders bedoeld hadden moet het gebouwde systeem worden aangepast. Dat is een iteratie over de life cycle van het projekt heen, zie Fig. 21.2. Deze vorm van itereren wordt gekenmerkt door ontevreden gebruikers, lange doorlooptijden en hoge kosten. Een iteratief ontwerpproces funktioneert alleen als tijdens het ontwerpen konsekwenties van eisen kunnen worden vastgesteld. Tijdens de ontwerpfase mogen gebruikers nog terugkomen op eerder genomen beslissingen, en met andere cijfers komen. Dat alles versnelt het ontwerpproces niet, maar beter nu enige vertraging, dan irritatie als het systeem is gebouwd. Iteratie voorkomt irritatie. Samengevat kunnen we het zo zeggen: gebruikers kunnen niet beoordelen wat ze niet gezien hebben, automatiseerders kunnen alleen iets laten zien dat ze gebouwd hebben. Deze situatie zorgt voor kommunikatieproblemen op allerlei gebied. De oplossing kan gevonden worden in het gebruik van methoden waarmee de gebruikers tijdens het

dagelijkse werk.
Als het systeem is ontworpen hoeven de gebruikers alleen nog maar te wachten op de overdracht. Dat wachten wordt meestal snel onaangenaam omdat de gebruiker het bouwproces niet kan beinvloeden, hij geen idee heeft van het werk van systeemontwerpers en programmeurs en hij bovendien regelmatig gekonfronteerd wordt met verschoven data: automatiseringsprojekten hebben nu eenmaal de neiging om uit te lopen. Voor een deel komt dat door planningsfouten van de automatiseerders, maar vaker door het krappe tijdschema dat het gebruikersmanagement heeft opgesteld. Daarbij laten we dan nog maar buiten beschouwing dat gebruikers soms tijdens de bouw nog met nieuwe voorstellen komen.
Het zal duidelijk zijn dat in de verschillende fasen de kommunikatie tussen gebruikers en automatiseerders verschillende onderwerpen betreft. In de analysefase gaat het om de bestaande situatie, kwalitatief en kwantitatief. Kwalitatief bezien, is eigenlijk het enige probleem de diepgang: hoe gedetailleerd moet de bestaande situatie in kaart worden gebracht. Dat is niet zozeer een probleem voor de gebruikers, alswel voor de informatie-analisten. De kwantitatieve aspekten leveren meestal meer problemen op voor de gebruikers. Informatie-analisten vragen soms naar cijfers waar een gebruiker nog nooit over heeft nagedacht. Als gebruikers niet inzien wat er gedaan wordt met de verstrekte cijfers zullen ze ofwel de informatie-analist een schatting laten doen of een getal noemen waarvan ze denken dat het geen negatieve gevolgen heeft voor hun werk, hun afdeling of hun koninkrijk. De terughoudendheid van sommige gebruikers bij het gesprek over kwantiteiten is verklaarbaar. In de praktijk is het meestal zo, dat ze cijfers mogen verstrekken in de analysefase en bij de overdracht merken waar dat toe geleid heeft, zonder dat ze precies begrijpen hoe.
Als tijdens het ontwerpproces methoden worden toegepast die cijfers van gebruikers vertalen naar gevolgen, zullen gebruikers tijdens de analysefase anders gaan reageren. In principe zijn de tijdens de analyse fase verstrekte cijfers vrijblijvend. Tijdens het ontwerpproces wordt de nieuwe situatie pas echt duidelijk voor de gebruikers. Dan mogen ze ook vragen verschillende situaties door te rekenen. Zeker als de bestaande situatie wezenlijk zal veranderen qua kwantiteiten, wordt het moeilijk om een cijfer te noemen. Automatiseerders, die beschikken over de juiste middelen kunnen gemakkelijk voor verschillende situaties de gevolgen bepalen. Uiteindelijk moet de gebruiker dan, het geheel overziend, èen keus doen. Het is misschien niet gemakkelijk om een beslissing te nemen, maar veel erger is het ergens voor te moeten kiezen, zonder de gevolgen te kunnen overzien.
Ontwerpen is een iteratief proces. Dat betekent dat er schattin-

ander te weten. Automatisering staat of valt immers met de bruikbaarheid ervan. Methoden die dienen om die bruikbaarheid door gebruikers te laten vastleggen moeten door elke manager op hun waarde beoordeeld kunnen worden.
In grote lijnen komt automatiseren voor de gebruikers neer op drie fasen: analyse, ontwerp en overdracht. In de analysefase wordt de bestaande situatie in kaart gebracht. In het algemeen zijn daarbij gebruikers betrokken van het type C of D. Zij kennen de processen, de gegevens en de knelpunten. Deze kommunikatie kost de gebruikers veel tijd en het is van belang dat ze zich dat realiseren, want kleine misverstanden in de analysefase kunnen grote gevolgend hebben voor het uiteindelijke systeem of voor de doorlooptijd van een projekt. Het belang van een nauwkeurige analyse wordt vaak onderschat. Vooral op managementniveau wordt er vaak alleen maar gekeken naar het moment waarop het systeem in bedrijf komt. Vaak denkt men op dat niveau dat de manier van werken binnen het bedrijf erg eenvoudig is, heel goed is vastgelegd of net zo is als in andere bedrijven. Het verschil tussen het procedurehandboek en de werkwijze op de werkvloer komt vaak pas bij een nauwkeurige analyse aan het licht.
Een heel ander probleem is de objektiviteit van de gegevens. Een ambitieuze verkoopleider noemt meestal een ander gemiddeld aantal orders per dag dan het hoofd van de orderadministratie.
De volgende fase is die van het ontwerp. Als het geautomatiseerde systeem wordt ontworpen op basis van de handmatige situatie, wordt hoogstens de bestaande situatie verbeterd: een aantal knelpunten verdwijnt, de hoeveelheid papier vermindert, de doorlooptijd wordt korter enzovoort. Wanneer echter gebruikers worden ingeschakeld bij het ontwerpproces kunnen nieuwe mogelijkheden naar voren komen. Zeker bij gebruikers van het type B en C komt het kreatieve meedenken pas op gang als ze op hun werkplek achter het beeldscherm zitten. Die situatie moet dus tijdens het ontwerpproces ontstaan. We zullen de methoden die daartoe een bijdrage leveren bespreken in de volgende hoofdstukken.
In het ontwerpproces is een aantal elementen te onderscheiden. De verwerking door de computer moet worden ontworpen. In grote lijnen is dat het proces van invoer, verwerking en uitvoer van gegevens. Daarnaast moeten de gegevens worden opgeslagen op een manier die past bij de verwerking. De struktuur van gegevens wordt in kaart gebracht tijdens de analysefase, tijdens de ontwerpfase wordt de opslag in het computergeheugen ontworpen. Bij interaktieve toepassingen moet dan de dialoog nog worden ontworpen. Voor gebruikers van het type A en B, die dagelijks met het beeldscherm moeten werken, is die dialoog het belangrijkste. Alle andere zaken zijn een keer ontworpen en liggen bovendien vast in de programmatuur. De procedure aan het beeldscherm bepaalt echter hun

praktijk moet dit uiteraard in verstandige banen worden geleid. Daarbij moet bedacht worden dat er naast technische ook funktionele beperkingen kunnen worden opgelegd. Zo zal er moeten worden vastgelegd wie de eigenaars van gegevens zijn en wie de gegevens beheert, wie ze mag wijzigen, wie ze alleen mag opvragen.
Het taktisch management dat verantwoordelijk is voor de keuze van methoden, dient zich dit soort zaken te realiseren. Afhankelijk van de situatie dient er dus een aantal zaken te worden geregeld, zoals bovenstaand aangegeven.
Automatiseerders worden opgeleid in het kommuniceren met gebruikers via methoden (16), maar ook de gebruikers moeten voorbereid worden op die kommunikatie. Niet door hem een kursus computertechniek te geven maar door hem te laten zien wat de mogelijkheden zijn van het werken met een computer via een beeldscherm. Hij hoeft niet te weten hoe een computer werkt maar hij mag best in een vroeg stadium weten dat er op zijn toekomstige beeldscherm slechts 24 regels tekst kunnen worden getoond van 80 tekens lang (15).
Het is van belang dat gebruikers op de juiste wijze worden voorbereid op een gekozen methode van ontwerpen.
Het is van taktisch belang zich te verdiepen in die aspekten van een systeemontwikkelingsmethode die voor gebruikers van belang zijn. Methoden in de ontwerpfase zijn van het grootste belang! Daar wordt vastgelegd hoe toekomstige systemen zich zullen gedragen, hoe ze bediend moeten worden en welke resultaten ze opleveren en tenslotte wordt nog gewezen op de dokumentatie die uit methoden te voorschijn komt.
Ontwerpspecifikaties in een onleesbaar schrift weergegeven zijn niet akseptabel. De meeste dokumenten van automatiseerders zijn onleesbaar voor gewone gebruikers. Met gewone gebruikers worden in dit geval gebruikers bedoeld van het type A, B en C. Wanneer de tijd daarvoor beschikbaar is en het opleidingsniveau van gebruikers ervoor geschikt is, kunnen gebruikers opgeleid worden in het maken en lezen van deze ontwerpdokumenten.
In veel bedrijven is aan een van deze twee voorwaarden niet voldaan. Dat betekent dat de dokumenten voor gewone gebruikers leesbaar moeten zijn. De transaktieschema's die in dit boek behandeld worden zijn daarvan goede voorbeelden. Er zijn bedrijven waar de gebruikers zelf transaktieschema's maken zonder enige opleiding.
Samengevat kunnen we vaststellen dat de situatie voor de gebruiker in meer dan een opzicht veranderd is. Het tijdsprobleem zal op creatieve doch effektieve wijze moeten worden opgelost. De andere zaken zijn opgelost met de keuze van de juiste methoden. Verdiep u in die methoden voor alle aspekten die iets met de gebruikers te maken hebben. Laat dat niet aan specialisten over: het taktisch management dient van deze aspekten zelf het een en

is er nu eenmaal niet voor de automatisering. Het "echte" werk gaat voor en dat is fnuikend voor een projektgroep. De vervanger is dan niet ingewerkt, heeft de stukken niet bestudeerd enzovoort. Wanneer de vergaderingen dan ook nog verzanden in ingewikkelde diskussies over entiteiten, processen, funkties, en andere vreemde onderwerpen, dan stuurt hij al gauw zijn vervanger.
Een derde probleem doet zich voor bij de keuze van de gebruikers die het gaan zeggen. Wie van de gebruikers worden de medeontwerpers? Gaan ze kommuniceren met een achterban? Neemt een klein selekt groepje beslissingen die jaren het werk van velen kunnen beinvloeden? Hoe voelen de geselekteerden zich daarbij? Wie wil de verantwoording dragen voor iets dat een ernorme investering betekent en dan achteraf toch niet blijkt te zijn wat andere gebruikers ervan verwacht hadden?
Kortom, de situatie van vroeger was verre van ideaal, maar deze aanpak is voor gebruikers minstens zo onprettig.
Het tijdsprobleem is niet met een methode op te lossen. Het is een essentieel probleem. Van het taktisch management mag verwacht worden dat ze dit soort problemen aan kan. Wanneer er gekozen is voor de nieuwe manier van automatiseren dan heeft dat een hoge prioriteit.
Zoals er bij ziekte maatregelen getroffen worden om de afdeling te laten doordraaien zo kan dat ook wanneer iemand betrokken is bij automatisering. Automatiseren is een kostbare aangelegenheid. Het is dus niet akseptabel dat gebruikers geen tijd krijgen om te specificeren wat er gebouwd moet worden. De tijd dat er bij slecht funktionerende systemen gewezen kan worden naar de automatiseerders is echt voorbij.
De andere genoemde problemen zijn simpel op te lossen. Bij dialoogsimulatie wordt met een portable microcomputer de werkpleksituatie gesimuleerd. Met andere woorden, het toekomstige werkplekbeeldscherm is nu het beeldscherm van een microcomputer. Alles wat later op een beeldscherm gebeurt wordt nu gesimuleerd op de microcomputer. De gebruiker ziet en ervaart nu konkreet wat een computer voor zijn werk betekent. Hij kan nu bonnen of fakturen of orders intypen zoals dat later zal gaan op het echte beeldscherm. Een ander aspekt van deze portable dialoogsimulator is dat hij overal mee naar toe genomen kan worden. Er is in principe geen beperking aan het aantal gebruikers dat er mee mag werken. Zelfs bij geografisch sterk verspreide vestigingen kan "on site" gewerkt worden met de toekomstige automatisering. Dat betekent dat het kleine geselekteerde groepje gebruikers van de projektgroep wat dit aspekt betreft ineens wordt uitgebreid met alle gebruikers. Dat betekent dat de verantwoordelijkheid voor het ontwerp gespreid wordt over de gehele groep. Iedereen kan kommentaar geven, voorstellen doen of alternatieven willen zien. In de

besprekingen wilde de kommunikatie niet erg vlotten (21).
De slotkonklusie van de gebruikers was vaak dat de automatiseerders hen niet begrepen en wat ze wel begrepen kon niet gerealiseerd worden of kostte te veel. Vanuit de ivoren toren werden systemen ontwikkeld en gebouwd. De gebruikers hoefden die alleen maar achteraf te beoordelen. Wanneer sommige delen niet bruikbaar waren of een opossing vormden voor een inmiddels niet meer bestaand probleem, dan was de kritiek eenvoudig en vernietigend. Achterafpraten is nu eenmaal eenvoudiger dan van te voren bedenken wat er fout kan gaan. Deze manier van werken had voor de gebruikers in ieder geval het voordeel dat de hele automatisering hun weinig tijd kostte. Informatie-analisten namen een paar interviews af en verder was er dan nog de akseptatietest. Meestal bleek dat een ander woord te zijn voor "invoering en gebruikerstraining".
Er is de laatste jaren met name onder de automatiseerders het een en ander veranderd. In veel opleidingen wordt gehamerd op samenwerking met gebruikers, want miljoenen investeren in systemen die in het gebruik niet voldoen, is wat moeilijk verteerbaar voor de meeste bedrijven. Automatisering komt al in de ontwerpfase naar de gebruikers toe, en er zijn methoden ontwikkeld om gebruikers mede-ontwerper te laten zijn.
Dat levert enkele problemen op. Het grootste is het feit dat het achterafpraten over een in gebruik zijnd systeem, vervangen is door wensen uitspreken. Er wordt van de gebruiker verwacht dat hij weet wat hij wil en "gebruikers die niet weten wat ze willen krijgen iets anders". Dat betekent uitspraken doen en gegevens verstrekken. Kon hij vroeger zeggen: "Ze begrijpen het toch niet", nu mag hij het zeggen en er wordt geduldig net zo lang gepraat tot alles duidelijk is. Zo duidelijk, dat wat hij gevraagd heeft ook gebouwd gaat worden, precies zoals hij het gespecificeerd heeft. Die akseptatietest krijgt nu wel een heel ander aksent! Daar wordt hem getoond wat hij destijds bestelde. Gebreken en fouten kunnen nu niet meer worden afgeschoven naar de automatiseerders: de gebruiker is koning en hij heeft het besteld. De automatiseerders voelen zich prima en de gebruikers hebben er een probleem bij.
Een tweede probleem dat zich voordoet is de beschikbare tijd. In de meeste bedrijven zijn de mensen volledig bezet. Automatisering is qua ontwerp een kortlopend projekt dat "erbij" gedaan moet worden. Dat wil zeggen dat iemand die eigenlijk toch al geen tijd had, wordt aangewezen als lid van een projektgroep. Voor noodsituaties wordt een vervanger aangewezen en een slimme vervanger zorgt zelf ook weer voor een vervanger. In deze gevallen blijken noodsituaties dagelijks voor te kunnen komen.
Het is natuurlijk een kwestie van prioriteiten, maar het bedrijf

ringskennis, dat ze bijna informatie-analist zijn. Het zijn de gebruikers die schema's van de informatie-analisten kunnen leren en hun jargon verstaan. Het zouden de beleidsbepalende gebruikers in (9) , blz. 471 kunnen zijn en ze kunnen behoren tot het taktisch gebruikersmanagement.
Er zijn in principe twee soorten informatie-analisten. De ene groep wordt gevormd door degenen die hun loopbaan zijn begonnen als automatiseerder en het bekende trajekt van operator via programmeur en systeemontwerper tot informatie-analist hebben afgelegd, of ze hebben een opleiding genoten waarmee ze meteen informatie-analist zijn geworden. De andere groep bestaat uit mensen die eerst een vak hebben geleerd en pas later zijn overgestapt naar de automatisering. Zij hebben dus een grote hoeveelheid materiedeskundigheid en zullen uitstekend kunnen kommuniceren met gebruikers. Een gevaar daarbij is dat de gebruikers de automatisering te veel aan hen overgelaten en zelf niet voldoende bijdragen in het ontwerpen van nieuwe systemen.
Beide groepen gaan verschillend om met de gegevens van de gebruikers. De eerste groep zal ze aksepteren zoals ze verstrekt worden, de tweede groep kan vaak beoordelen of de gegevens realistisch zijn. In het eerste geval kan hoogstens uit de konsekwenties blijken of de cijfers kloppen met de werkelijkheid. Soms komen die gevolgen pas naar voren tijdens het technisch ontwerp en dan is het nog de vraag of ze zijn te herleiden tot gegevens die verstrekt werden door bepaalde gebruikers. De eerste groep is in het algemeen beter in het beoordelen van de technische mogelijkheden. Ze kennen de mogelijkheden en beperkingen van computersystemen beter dan de tweede groep.
Als we de informatie-analist even beschouwen als de kontaktpersoon tussen gebruikers en automatiseerders, zal de eerste groep beter kommuniceren met de automatiseerders, de tweede groep beter met de gebruikers.
De verschillen tussen mensen, tussen bedrijven en tussen projekten vragen om een vaste manier van werken, die gezamelijk is gekozen en gebruikers de kans geeft vat te krijgen op de automatisering.

21.3 Kommuniceren met automatiseerders

Automatisering begint al geschiedenis te krijgen, de visie van gebruikers op die automatisering ook.
Reeds twintig jaar geleden beklaagden gebruikers zich over de kommunikatie met automatiseerders. Even afgezien van het verschil in taal en schrift, viel er ook funktioneel weinig met hen aan te vangen. De automatiseerders werkten sowieso al vanuit hun ivoren toren en daarom was er al een barriere geschapen. Maar zelfs op

matie verder verwerkt wordt valt buiten hun gezichtsveld. In de geautomatiseerde situatie zouden we kunnen denken aan data entry-typistes die snel en liefst foutloos moeten intypen, maar geen idee hebben wat er met de gegevens gebeurt. In het algemeen hebben ze geen kennis van de professionele automatisering. Ze zullen van de hele automatisering slechts het beeldscherm en het toetsenbord goed kennen. De dialoog met de computer is voor het van groot belang en enig belang. Ze worden meestal nauwelijks bij de automatisering betrokken en merken pas wat een beeldscherm voor hun werk betekent als het systeem is gebouwd. Het zijn de operationele gebruikers in (9), blz. 471.

- Gebruikers van het type B. Deze groep in algemene termen beschrijven is al moeilijker. In de praktijk zijn het vaak degenen die leiding geven aan een groep gebruikers van het type A. Zij weten waarom dokumenten gekontroleerd worden, welke kontroles het belangrijkste zijn en waarom. Vaak zijn ze zelf voortgekomen uit die groep op grond van ervaring en aantal dienstjaren. Soms hebben ze enige belangstelling voor de automatisering, al zou het alleen al zijn uit hoofde van hun funktie. Ze zijn in ieder geval geinteresseerd in de dialoog met de computer, maar zullen ook vragen naar het waarom van die dialoog. In dat geval zullen ze ook komen met voorstellen ten aanzien van nieuwe toepassingen. Dat laatste kan bij de gebruikers van het type A natuurlijk ook, maar die voorstellen moeten meestal toch beoordeeld worden door gebruikers van type B. Gebruikers van het type A en B worden vaak eindgebruikers genoemd.

- Gebruikers van het type C. Deze groep behoort tot materiedeskundigen: zij beheersen het hele vakgebied. In een produktie-omgeving is dat de bedrijfsleider, in de administratie de hoofdboekhouder, bij de verkoopfunktie de salesmanager. Vaak zijn het dus managers van bedrijfsonderdelen, soms zijn het staffunktionarissen van een bedrijfsonderdeel. Zij kennen alle bedrijfsprocessen en alle gegevens en gegevensstromen, evenals de relatie met andere bedrijfsonderdelen. Zij zijn degenen met wie de informatie-analisten de processen en gegevens in kaart brengen. Zij specificeren de funkties die het informatiesysteem moet vervullen. Voor hen is de dialoog met de computer van ondergeschikt belang omdat ze zelf niet of nauwelijks met het beeldscherm zullen werken. Zij besturen de bedrijfsprocessen, maar voeren ze niet zelf uit. Hoogstens gebruiken zij het beeldscherm voor het verkrijgen van managementinformatie over de bedrijfsprocessen. Het dialoog-ontwerp ten dienste van bedrijfsprocessen zullen ze overlaten aan gebruikers van het type B. Het zijn de operationele gebruikers in (9), blz. 471.

- Gebruikers van het type D. Zij beschikken over de vakkennis van gebruikers van het type C en hebben daarnaast zoveel automatise-

gereedschap is om een schermlay-out te maken, maar gebruikers weten niet wat een schermlay-out is, dan is het gereedschap niet optimaal te gebruiken. Dan maakt de automatiseerder wel uit wat de beste lay-out is. Bij projektaanpak gaat het dan om de manier van werken. Hebben de gebruikers op het juiste moment de juiste gegevens beschikbaar? Weten ze wanneer de bouw begint en er dus een eind is gekomen aan de inspraak?
Bij het management gaat het dan om het beheer en de besturing van de gebruikers in het kader van alle genoemde aspekten. Daar gaat het bijvoorbeeld om het bekende tijdprobleem: is er inderdaad voor gezorgd dat op het juiste moment de juiste mensen beschikbaar zijn?
Zoals blijkt uit de formule heeft de faktor management een enorme invloed. Op de werkvloer kan men nog zo gemotiveerd zijn en nog zo ter zake kundig, wanneer het management zijn zaken niet regelt, door bijvoorbeeld onvoldoende tijd beschikbaar te stellen, dan is dat alles te vergeefs. Verder is de invloed van M1 veel groter dan die van M2 en M3. Daarmee is aangegeven dat methoden en gereedschappen op zich het succes niet garanderen. Wanneer ze echter niet gedefinieerd zijn, moet iedereen maar afwachten wat het vakmanschap tot stand zal brengen.
Samenvattend kunnen we zeggen, dat automatisering waarschijnlijk nooit volmaakt zal worden omdat we allemaal maar mensen zijn. Wanneer aan een aantal voorwaarden is voldaan, kunnen we best een goed systeem bouwen.

21.2 Wie zijn de gebruikers, wie de informatie-analisten?

Elke indeling van mensen roept vragen op. Ten eerste omdat het riekt naar het verheffen van de een boven de ander, ten tweede omdat er geen twee bedrijven hetzelfde zijn. De indeling die wij nu zullen maken heeft niets te maken met niveauverschillen. Alle mensen zijn gelijkwaardig, ze zijn gelukkig niet gelijk. Niet iedereen heeft dezelfde opleiding, dezelfde verantwoordelijkheid of dezelfde belangstelling. De indeling heeft te maken met de toepassing van de te bespreken methoden en de kommunikatie met de automatiseerders.
We zullen vier typen gebruikers onderscheiden en per type letten op hun inzicht in het bedrijfsproces waarbij ze betrokken zijn, in wat een computer voor hun werk kan betekenen en in de daarvan afgeleide mogelijkheid om op te treden als mede-ontwerper.
- Gebruikers van het type A. Hiermee wordt in de niet geautomatiseerde situatie, de groep bedoeld die bijvoorbeeld eenvoudige administratieve handelingen verricht zonder verantwoordelijk te zijn voor wat er verder gebeurt. Ze sorteren bijvoorbeeld dokumenten en kontroleren of ze volledig zijn ingevuld. Hoe de infor-

In deze formule wordt daarmee bedoeld de projektaanpak voor de automatiseerders. Gebruikers verdiepen zich meestal niet in de methoden die automatiseerders gebruiken om bijvoorbeeld hun dokumentatie op te zetten of hun programma's te maken. Toch merken gebruikers best of er aan automatiseringszijde projektmatig wordt gewerkt. Al zou het alleen maar zijn dat ze altijd gekonfronteerd worden met dezelfde dokumenten en terminologie onafhankelijk van de persoon van de automatiseerder.Daarnaast worden ze, zoals in dit boek is aangegeven, steeds op het juiste moment ingeschakeld door automatiseerders.
Het vakmanschap van automatiseerders (M1) wordt door gebruikers beoordeeld met een cijfer van 0 - 10. Daarbij gaat het om aspekten die een gebruiker kan beoordelen. Vakmanschap ten aanzien van omgaan met gebruikers, openstaan voor voorstellen van gebruikers, gebruikerswensen vastleggen op een voor gebruikers leesbare wijze. Maar daarnaast ook vakmanschap in het ontwerpen van gebruikersvriendelijke systemen. Of daarbij methoden worden toegepast is al heel gemakkelijk te beoordelen. Wanneer iedere automatiseerder zijn eigen methode blijkt te hebben dan is het slecht gesteld met de methoden. Bij middelen (M3) gaat het om gereedschappen waarmee de gebruiker in aanraking komt en die hij kan beoordelen op gebruik, mogelijkheden en dergelijke. Het deel van de formule tussen de haken gaat over het vakmanschap, de methoden en de gereedschappen die gebruikt worden. Het tweede deel gaat over het kader waarin het vakmanschap moet opereren: projektaanpak en management. Het cijfer voor projektaanpak is in de praktijk natuurlijk nooit nul. Ook al is er geen geformaliseerde aanpak, er wordt gewerkt en dus op een bepaalde manier. Het laagste cijfer voor projektaanpak is dus een 1. Maar zelfs al is het cijfer voor projektaanpak een tien, bij afwezig management, een nul dus, het resultaat blijft een 1 voor het tweede deel van de formule.
De formule zou ook gebruikt kunnen worden om de gebruikerssituatie eens te beoordelen. Het gaat dan uiteraard om haar vakmanschap en management om de automatisering tot een succes te maken. Bij vakmanschap gaat het dan bijvoorbeeld niet zozeer om de materiedeskundigheid alswel om de materie in kaart te brengen. In hoeverre kennen gebruikers hun eigen bedrijf werkelijk? Is de manier van werken eigenlijk wel ergens vastgelegd? Hoe groot is het verschil tussen het procedurehandboek en de gang van zaken op de werkvloer?
Bij methoden gaat het dan om de aansluiting op de methoden die de automatiseerders toepassen om gebruikerseisen zo goed mogelijk in kaart te brengen. Als er een methode is om responsetijden voelbaar te maken maar gebruikers weten niet wat responsetijden zijn, dan zijn ze onvoldoende voorbereid op samenwerking met automatiseerders. Hetzelfde geldt voor de gereedschappen. Wanneer er een

analyse is een begrotingsmethode die bepaalde kengetallen oplevert. Het rekenkundige model van een transaktie, gevat in een computerprogramma. Het transaktieschema, zoals dat samen met gebruikers is ingevuld, wordt door transaktie-analisten vertaald naar een transaktiedetailschema. Op dat detailschema worden alle kwantitatieve gebruikersgegevens vastgelegd. Dat detailschema is input voor het rekenprogramma. De resultaten van dat rekenprogramma zijn gegevens voor de automatiseerders voor het maken van o.a. een kostenplaatje van het netwerk.
De laatste van de 4 M`s is die van het management. Dat betekent in het kader van het onderwerp: management over de andere 3 M`s. Er moet bij het uitvoerend gebruikersmanagement precies bekend zijn welke methoden er gekozen zijn, welke dokumenten daarbij voor hen van belang zijn, welke middelen er beschikbaar zijn. In dit verband is het van belang dat het uitvoerend management goed is voorbereid op deze taak. Management zonder voldoende kennis van zaken blijkt in de praktijk meestal niet te werken. Waar het erom gaat gebruikers in te schakelen bij het ontwerp van hun nieuwe hulpmiddelen moet elke kans op uitglijden voorkomen worden. Met name het uitvoerend gebruikersmanagement moet goed ingevoerd zijn in de gebruikersaspekten van de projektaanpak. Vaak valt echter deze groep tussen de wal en het schip omdat ze het wel begrijpen en `s avonds wel eens wat lezen over automatisering. De opleiding van het uitvoerend gebruikersmanagement moet duidelijk zijn omschreven en als taak zijn toegewezen aan bijvoorbeeld een opleidingsfunktionaris. De komplete beschrijving van inhoud van deze opleiding valt buiten het kader van dit boek. Diverse opleidingsinstituten bieden een adequate invulling. Van belang daarbij is dat de opleiding moet kunnen wordenaangepast aan de bedrijfssituatie qua projektaanpak, beschikbare methoden en middelen. Deze paragraaf zou in de vorm van een formule kunnen worden samengevat:

$$PR = (M1 + M1 \times M2 + M1 \times M2 \times M3)\ PA^{M4}$$

PR: Het projektresultaat
M1: Het vakmanschap
M2: De methoden
M3: De middelen
M4: Het management
PA: De projektaanpak

De resultaatformule

In de formule komen de 4 M`s voor tezamen met de projektaanpak.

analysis for data transmission" dat hem in de praktijk steeds weer is gebleken dat 80% van de problemen rond netwerken, performance en responsetijden voorkomen had kunnen worden door enig rekenwerk vooraf.
Transaktie analyse kan gezien worden als een begrotingsmethode. Begrotingen kunnen varieren van ruime schattingen tot nauwkeurige analyses. Transaktie analyse kan zowel gebruikt worden in een vroegtijdig stadium van een projekt, maar dan heel grof, als zeer gedetailleerd tijdens het technisch ontwerp. Zoals voor alle begrotingsmethoden, geldt ook voor Transaktie analyse: garbage in, garbage out. De resultaten van Transaktie analyse zullen voor zover van belang voor het taktisch management dan ook nog verder worden besproken. De kosten van de uitvoering van Transaktie analyse zijn verwaarloosbaar ten opzichte van de totale kosten van een projekt.
Het is van taktisch belang ervoor te zorgen dat gebruikers zo snel mogelijk gekonfronteerd worden met de konsekwenties van hun eisen. Evenzo is het van belang om in probleemsituaties snel vast te kunnen stellen waar de knelpunten zich bevinden. Daarom is het voor het taktisch management van belang de huidige gang van zaken in de automatisering te evalueren en zich te verdiepen in methoden die een bijdrage kunnen leveren ter verbetering van de situatie.
Bij methoden horen vaak de passende gereedschappen, terwille van de alliteratie hier middelen genoemd.
Bij dialoogsimulatie is dat een dialoogsimulator. Voor de oppervlakkige toeschouwer is een dialoogsimulator bijna hetzelfde als een "screenpainter". Bij screenpainting wordt met behulp van het toetsenbord een schermlay-out gemaakt en getoond aan de gebruiker. Dan kan er alleen gediskussieerd worden over de schermindeling. Bij dialoogsimulatie gaat het om veel belangrijker aspekten.
Bij dialoogsimulatie "werkt" de dialoog. Dat wil zeggen: de gebruiker kan gegevens intypen en het systeem reageert met het displayen van gegevens eventueel na verloop van een ingestelde response tijd. Wanneer een transaktie is opgebouwd rond een aantal verschillende schermen dan volgen tijdens de dialoog-simulatie die schermen elkaar op zoals dat in werkelijkheid ook gebeurt, eventueel op basis van bepaalde keuzekriteria. Kortom de gebruiker kan heel konkreet zijn toekomstige transakties uitvoeren. Hij ziet hoe de dialoog met de computer aansluit bij de overige handmatige handelingen. Vaak lijkt een ontworpen dialoog op het eerste gezicht bruikbaar maar na verloop van tijd komen de bezwaren naar voren. Dat effekt kan nu al in de ontwerpfase worden bereikt.
Het middel bij Transaktie analyse is het rekenmodel. Transaktie

TRANSAKTIESCHEMA centraal

Menselijke handelingen en bewerkingen	Transport	Machinale verwerking
- lezen - opzoeken - kontroleren - zich verplaatsen - schrijven - praten		
met als laatste:		
- intypen van gegevens	----)	Wat moet de computer doen: - opzoeken - vergelijken - berekenen
Lezen, ... intypen	(----	- displayen
	----)	------- -------
Lezen, ... intypen	(----	Displayen
	----)	------- -------
	(----	Displayen
Terug naar begin van transaktie		

Fig. 21.1 Wat staat er op een transaktieschema.

Hoe fraai het ook klinkt, deze samenwerking gaat niet vanzelf. Zelfs al zijn automatiseerder en gebruiker nog zo enthousiast er zijn konkrete methoden nodig omhet geheel overzichtelijk en dus bestuurbaar te houden. Het gaat immers meestal niet om een gebruiker maar om hele gebruikersgroepen. Het gaat niet om een getalenteerde informatie-analist. Daarnaast gebeurt alles in het bedrijf in het spanningsveld van tijd en geld . Gebruikers kunnen niet blijven komen met voorstellen. Sommige transakties zijn met een kleine aanpassing ook goed te gebruiken op andere afdelingen. Kortom, we zijn natuurlijk niet in een paradijs terecht gekomen van: "u vraagt, wij draaien". Dat betekent meestal afwegen, kiezen, prioriteiten stellen en technische beperkingen aksepteren. Dat kan alleen wanneer er volgens vaste methoden, met standaard dokumenten wordt gewerkt.
In het kader van dit boek gaat het daarbij uiteraard om die methoden die leiden tot het ontwerp van beeldschermtoepassingen al dan niet gebruikmakend van een netwerk. Daarbij is hetuitgangspunt dat de start van alle eisen bij de gebruiker ligt. Het ontwerpen van transakties op transaktieschema's en het evalueren van die ontwerpen met behulp van dialoogsimulatie worden beschouwd als de methoden om die gebruikers eisen vast te leggen. Transaktie analyse is een methode om die eisen te kwantificeren en te vertalen naar kengetallen voor terminalpark en netwerk ontwerp. Het transaktieschema is startdokument voor Transaktie analyse. Op die wijze worden dialoogsimulatie en Transaktie analyse gekoppeld tot transaktie-ontwerp. Naast gegevens voor terminalpark en netwerk levert Transaktie analyse ook kengetallen op om per transaktie de belasting op het systeem vast te stellen. Kortom, met behulp van Transaktie analyse worden on-line-transakties naar een aantal gezichtspunten kwantitatief in kaart gebracht. En dat is een vergeten terrein op de meeste automatiseringsafdelingen. Er wordt ontworpen en in produktie gebracht tot de responsetijden onakseptabel zijn geworden. Dan kan het wel een jaar duren voor een groter systeem is geinstalleerd. Daarnaast blijkt dat sommige systemen vanaf het begin onakseptabel werken.
Analyse van de transakties toont meestal direkt aan waar het knelpunt zit en dat het ontstaan ervan eigenlijk te voorzien was. Bijna alle ontwerpers denken te weinig kwantitatief. De logika van de programma's is meestal uitstekend. Achteraf blijken er vaak wat onvoorziene performanceproblemen op te duiken. Wanneer die alleen kunnen worden opgelost door de programmatuur aan te passen, betekent dat het zoveelste projekt waarvan de kosten hoger zijn dan verwacht.
De kracht van begrotende methoden als Transaktie analyse is dat er in de ontwerpfase al gewezen kan worden op konsekwenties van bepaalde ontwerpen. James Martin schrijft in zijn boek "Systems

nieuwe gereedschap zo effektief mogelijk te laten funtioneren. Daarbij gaat het om het ontwerpen van transakties. Op een transaktieschema wordt de hele procedure in kaart gebracht. Het geheel van menselijke handelingen tezamen met het werken met het beeldscherm. Zie Fig. 21.1. Op die manier is het voor de gebruiker van het begin af aan duidelijk wat de volgorde van werken is en wat de funktie van bepaalde dokumenten is. De aldus beschreven transaktie kan nu met behulp van dialoogsimulatie "live" uitgeprobeerd worden. Dan ervaart de gebruiker hoe het beeldscherm wordt ingepast in het geheel van handelingen. Daar is nog velerlei kommentaar mogelijk. Het belangrijkste is wel dat er alternatieven kunnen worden uitgeprobeerd. In het beste geval komt de gebruiker zelfs met voorstellen voor nieuwe transakties.
Er wordt veel geschreven over de sociale aspekten van automatisering. Verhalen over chips en werkeloosheid vliegen ons om de oren. Veel belangrijker zijn konkrete situaties van gebruikers. De computer kan zeker een deel van het werk overnemen. Wanneer echter het vakmanschap van de gebruiker op boven omschreven wijze wordt gekoppeld aan dat van de automatiseerder kan een geweldig stuk funktieverrijking ontstaan. De gemiddelde automatiseerder weet best hoe in het algemeen de voorraadadministratie geautomatiseerd moet worden.
In dat soort situaties hebben de automatiseerders voldoende knowhow. Heel anders wordt het in de specifieke werkomgeving van talloze werknemers. Daar zullen de ideeen over het gebruik van de computer voor een belangrijk deel bij de gebruikers vandaan moeten komen. Wanneer ze op de juiste wijze betrokken worden bij het ontwerpen van transakties zal de visie op de automatisering totaal veranderen. Transaktieschema's kunnen door gebruiker zelf ingevuld worden. Er zijn bedrijven waar de gebruikers, als ze met nieuwe wensen naar voren komen, te horen krijgen: maak maar een transaktieschema. Op basis van zo'n eerste aftrap komt de diskussie snel en effektief tot stand. Naarmate gebruikers gaan ervaren hoe konkreet nieuwe voorstellen kunnen worden uitgetest, aangepast en misschien wel van tafel worden geveegd, hoe meer ze geneigd zijn creatief over hun funktie te gaan nadenken. Natuurlijk moet zich in de praktijk een evenwicht instellen tussen enthousiasme en begrip van kosten. Dat geldt echter in veel bedrijfssituaties. Vaak kunnen voorstellen van medewerkers niet gehonoreerd worden omdat ze te weinig opleveren en te veel kosten. Daarmee zijn ideeënbussen toch niet onbruikbaar geworden? Ideeën over funktieverrijking zullen in principe uit de gebruikerswereld moeten komen. De gebruikers kennen hun vak. De timmerman gaat ontdekken dat dank zij moderne lijmtechnieken ineens bepaalde konstrukties mogelijk worden.
Dat brengt ons bij de tweede M.

tuur. De faciliteiten die de apparatuur biedt, moeten passen bij de funktie die verricht wordt op die werkplek. Om een funktie te kunnen uitvoeren, bijvoorbeeld via het beeldscherm, zullen een aantal transakties mogelijk moeten zijn. Voorbeelden van transakties op een werkplek op een verkoopafdeling kunnen bijvoorbeeld zijn: het invoeren van een order, het opvragen van een overzicht van de voorraad, het opvragen van lopende orders, het wijzigen van orders. Zo'n transaktie bestaat in de ogen van automatiseerders uit afwisselend intypen en lezen op het scherm.
Voor de medewerker bestaat het uit bijvoorbeeld het aannemen van de telefoon, het kontakt maken met de klant, het vaststellen van het doel van het gesprek, het intypen van een keuze op een menuscherm enzovoort. Kortom, meestal is een transaktie veel meer dan intypen en lezen. Vaak is het werken met het beeldscherm maar een onderdeel van een transaktie. Zeker als er nog gemanipuleerd moet worden met allerlei dokumenten.
Voor automatiseerders betekent het automatiseren van een transaktie dan ook bijna niets meer dan het maken van een beeldschermlayout. Op zo'n lay-out kan een automatiseerder zien wat de gebruiker moet intypen en wat de computer op het scherm moet zetten en waar. Daarmee is de zaak geautomatiseerd. En of de gebruiker dan maar even z'n handtekening wil zetten onder dat dokument. Dat is de antieke manier van automatiseren. Vroeger was het enige wat een gebruiker van de computer zag, een pak papier.
De lay-out van die dokumenten was vaak wel met hem afgestemd. Veel automatiseerders maken nu in plaats van een dokument-lay-out een beeldschermlay-out en daarmee is de zaak beklonken.
De gebruiker aksepteert die lay-out. Daarmee is vastgesteld wanneer wat moet worden ingetypt en wanneer en waar de computer iets op het scherm zet. Kortom, daarmee is de dialoog met de computer vastgelegd. Het zal duidelijk zijn dat de gemiddelde gebruiker zich op basis van zo'n dokument niet voor kan stellen hoe een beeldscherm nu funktioneert binnen de transaktie. Immers, zoals gezegd, vaak is de dialoog met de computer maar een onderdeel van de hele transaktie.
Van de huidige handmatige situatie komen misschien een aantal dokumenten te vervallen de overige moeten aansluiten op het werken met het beeldscherm. Kort gezegd, er moeten eerst transakties ontworpen worden.
Een medewerker op een bepaalde werkplek vervult daar een funktie. Om die funktie te vervullen zal hij een aantal aktiviteiten uitvoeren. Stel dat een deel van die aktiviteiten voor automatisering in aanmerking komt. Per aktiviteit kunnen een of meer transakties nodig zijn. Dat betekent dat, wanneer hij via een goede opleiding is voorbereid op het werken met beeldschermen, hij nu zijn vakmanschap op creatieve wijze moet inzetten om dit

Hoofdstuk 21
Mensen, methoden en middelen

21.1 Vier × M

Mensen, methoden, middelen en management. De volgorde is wel van belang, maar management had ook voorop mogen staan.
Eerst de mensen. In dit geval wordt er met mensen eigenlijk gedoeld op hun vakmanschap. Het vakmanschap van een timmerman is zijn vaktechnische kennis gekoppeld aan een grote dosis ervaring. Hoewel het gereedschap van de timmerman de laatste twintig jaar wezenlijk veranderd is, twijfelt er nog steeds niemand aan het vakmanschap van zo iemand. Ook al gebruikt hij totaal andere verbindingstechnieken dan vroeger, er blijft vakmanschap voor nodig om een wandmeubel te konstrueren.
Kortom, er is het een en ander veranderd aan de manier van werken, toch is hij vakman gebleven. Zo geldt dat ook voor medewerkers in het bedrijfsleven. Hun werkplek gaat er anders uit zien. De manier van werken wordt anders. Bepaalde dokumenten zullen uit de organisatie verdwijnen. Bij dat alles is slechts een ding van bebelang en dat is dat de vakman zelf betrokken is bij de verandering van werkmethoden.
Een werkplek is bedoeld voor een bepaalde funktionaris of medewerker. Een werkplek zal in de handmatige situatie vaak uit een bureau bestaan. De meeste funkties kunnen dan ook aan een willekeurig bureau worden uitgevoerd. In de geautomatiseerde situatie wordt de werkplek voorzien van meer of minder kostbare appara-

Deel 2

voor het taktisch gebruikersmanagement

Automatisering een zaak van automatiseerders?
Geautomatiseerd zijn in ieder geval niet!

moed ligt voor veel managers waarschijnlijk dan ook meer op het niveau van de prioriteiten in hun eigen agenda.

ment de vinger op de zere plek te leggen. Bij de automatisering van de gegevensverwerking kost dat al goud, bij de kantoorautomatisering in de informatiemaatschappij van morgen zal het desastreus blijken te zijn.

12.6 Strategische moed

Het is gebruikelijk dat op taktisch en uitvoerend niveau in probleemsituaties wordt vastgesteld, dat er geen beleid is. In veel gevallen is dat verwijt terecht. Op strategisch niveau beklaagt men zich vaak over te weinig beleidsondersteunende informatie uit de onderliggende niveau's.
Het is vaak al een probleem beleidsuitspraken te formuleren, maar er is altijd moed voor nodig om ze te doen. Beleid vaststellen betekent bijna altijd keuzes maken, zich vastleggen, zich beperken. We zullen ons nu bepalen tot enkele beslissingen ten aanzien van de automatisering, waar moed voor nodig is, maar die heilzaam zullen werken voor automatiseringsprojekten.
De eerste beslissing is om na het vooronderzoek geen opleveringsdatum meer te noemen. Dat kan een politieke storm betekenen bij vestigingsdirekteuren, taktische managers of afdelingschefs. Die stormen duren echter maar tot het logisch ontwerp van het eerste projekt of deel ervan.
De twee beslissing is dat iedere strategische manager zich zover in de automatisering gaat verdiepen dat men een presentatie van een uur kan houden rond Fig. 11.1. Tijdens die presentatie wordt onder andere duidelijk waarom er geen opleveringsdatum genoemd wordt. Dat is iets anders dan zich verdiepen in burgerinformatica, maar wel effektiever voor de beschreven problematiek. Het lezen van deel 2, voor het taktisch gebruikersmanagement vormt voldoende basis voor het houden van die presentatie. Natuurlijk zal niemand dat gaan doen. Toch is het nuttig eens goed na te denken over de werkelijke reden. Er worden wel eens algemene uitspraken gedaan over gebrek aan automatiseringskennis bij het management (40). Als geen enkel direktielid een konkreet idee heeft over de aanpak van de automatisering zullen in de toekomst, bij kantoorautomatisering, de problemen alleen maar groter worden dan ze nu al zijn.
De derde beslissing betreft het dwingend voorschrijven van het gebruik van methoden aan gebruikers en automatiseerders voor hun onderlinge kommunikatie.
Bij de vierde beslissing gaat het om het op schrift stellen van een aantal beleidspunten, minstens voor die gebieden die iets met automatisering te maken hebben.
De tijd die nodig is om deze beslissingen te nemen en de erbij behorende aktiviteiten uit te voeren is te overzien. De benodigde

Veel strategische vergissingen betreffen de kennis van het eigen bedrijf. Men denkt bijvoorbeeld te weten hoe de bedrijfsprocessen verlopen, maar kent niet het verschil tussen het procedurehandboek en de gang van zaken op de werkvloer. Veel bedrijven lopen goed omdat er op de werkvloer effektiever wordt gewerkt dan in de handboeken wordt beschreven. Sommige direkteuren denken dat hun bedrijf zo eenvoudig in elkaar zit, dat er helemaal geen handboeken nodig zijn. Zulke simpele bedrijfsprocessen zijn natuurlijk zeer goedkoop en vooral snel te automatiseren!
Anderen denken dat op uitvoerend niveau zoveel bekwaamheid aanwezig is, dat er niets beschreven hoeft te worden. Hoe onbeschreven situaties met een computer moeten worden geautomatiseerd is voor hen niet van belang, dat is een zaak van automatiseerders.
Een andere veel voorkomende vergissing betreft de oplossing van organisatorische, funktionele of kommunikatieve problemen. Automatisering lost, in z'n algemeenheid, die problemen niet op. Pas als de bestaande situatie met voldoende nauwkeurigheid is geanalyseerd en in kaart gebracht, kan erover gedacht worden een informatieverwerkend systeem zodanig te ontwerpen dat er een oplossing ontstaat voor een deel van de vastgestelde knelpunten. Er is dus een nauwkeurige, tijdvergende analyse nodig en een dito ontwerp. Dat geldt zelfs voor de zogenaamde eenvoudige, kleine bedrijven.
Uit een onderzoek (42) is gebleken dat de helft van de onderzochte bedrijven geen op schrift gesteld bedrijfsbeleid hebben. Als reden wordt opgegeven: daar hebben we geen tijd voor, daar zijn we een te dynamisch bedrijf voor, we passen ons aan aan de omstandigheden of iets dergelijks. Overtuigend klinkt het niet. De vraag is eerder of men er wel toe in staat is. Hoeveel direkteuren zouden wel eens een beleidsplan hebben gezien? Geen beleidsplan betekent meestal: geen informatieplan, dus ook geen automatiseringsplan en dus blijven we achter de problemen aan lopen: oplossingen ad-hoc, budgetten ad-hoc.
Het is een strategische vergissing de kosten/baten-analyse te baseren op de aanschaf van systemen en de besparing aan personeel, daarbij te lichtvaardig geloof te hechten aan uitspraken van computerleveranciers over het aantal mensen dat nodig is om hun systemen "in de lucht te houden". Enige kennis van zaken rond het managen van datacenters kan daarbij geen kwaad (25).
Ir. F.J. Philips hield eens de bouw van een kantoor aan de Bosdijk in Eindhoven tegen omdat hij het niet eens was met de verhouding tussen de oppervlakte voor werkruimte en die voor gangen, hallen en entree. Daarmee werd de heer Philips geen architekt, maar hij bemoeide zich op het juiste moment met de juiste details. Het is te vrezen dat het strategisch management in veel bedrijven te weinig kennis van zaken heeft om op het juiste mo-

geen zaak blijven van automatiseerders alleen. Zij bouwen het systeem, maar gebruikers ontwerpen het. Het is een vergissing te denken dat automatisering een zaak van interne of externe automatiseerders is. Het is ook een vergissing te denken dat men zich daarom op strategisch niveau moet bezig houden met burgerinformatica, methodologische informatica of technieken voor informatieverzorging. Op strategisch niveau dient men er voor te zorgen dat de kommunikatie tussen gebruikers en automatiseerders gemanaged kan worden. Dat is niet te regelen met het uitspreken van vertrouwen in interne of externe automatiseerders. Slechts de keuze van de aanpak kan overgelaten worden aan het taktisch management. Het is een strategische beslissing te denken dat men automatisering kan uitbesteden aan een bekend, vertrouwd en deskundig bureau. De kommunikatie tussen gebruikers en automatiseerders is in zo'n geval net zo belangrijk als bij uitvoering door eigen automatiseringspersoneel, het managen van die kommunikatie nog veel belangrijker! Men kan het werk uitbesteden, het beheer van de kwaliteit van het te ontwerpen produkt nooit. Daarom zijn te managen methoden die tijdens het ontwerp bij alle gebruikers zekerheid geven over het te leveren produkt.
Het is een vergissing te denken dat na een globale kosten/baten-analyse en een GO beslissing, de vraag gesteld moet worden wanneer het systeem zal worden opgeleverd of zal zijn ingevoerd. Als er namelijk een datum genoemd wordt, gaat die een eigen leven leiden. Automatiseringsprojekten hebben de naam uit te lopen, maar in de praktijk blijkt vaak dat de opleveringsdatum op een verkeerd moment is geschat. Na een vooronderzoek en een masterplan valt er nog niets te schatten. Kosten/baten-analyses tijdens het vooronderzoek zijn dan ook niet erg zinvol. De GO beslissing na het vooronderzoek moet een logisch ontwerp betreffen, waarbij gebruikers de toekomstige situatie volledig overzien zowel kwalitatief, kwantitatief als organisatorisch. Dan kan een voorlopige opleveringsdatum worden bepaald. Aan het eind van technisch ontwerp wordt door de betrokken gebruikers en automatiseerders vastgesteld of er aan de eisen, tijdens het logisch ontwerp gesteld wordt voldaan. Pas dan wordt een definitieve opleveringsdatum bepaald.
Het is een strategische vergissing te denken dat het vasthouden aan een genoemde en te vroeg vastgestelde opleveringsdatum van strategisch belang is. Het kan lijken alsof er enige maanden uitstel wordt voorkomen, maar de invoering kan een jaar vertraagd worden zonder dat enig strateeg daar nog iets aan kan veranderen. Tijdsdruk is de grootste vijand van elk projekt. Het eerste waarop dan bezuinigd wordt, is de kommunikatie met de gebruiker en dat is nu net de oorzaak van veel van de eerder genoemde problemen.

zijn.

- De automatiseringsafdeling ziet om uiteenlopende redenen geen kans zich aan die tijdsdruk te ontworstelen.
- Er is daardoor geen tijd voor een gedegen ontwerp en goede afstemming met de toekomstige gebruikers.
- Als het systeem wordt opgeleverd blijkt het zo slecht te werken dat iedereen zich vertwijfeld afvraagt hoe zo iets gebouwd kon worden.
- Die klachten komen terecht op het strategisch niveau waar men zich van geen kwaad bewust is en op zoek gaat naar de schuldigen op de onderliggende niveaus of een externe adviseur inschakelt.

Strategie heeft echter iets te maken met inzicht in zaken op lange termijn. Het ontwerp en de bouw van informatiesystemen kost veel meer tijd dan alles wat er aan vooraf gaat. Dat geldt zelfs wanneer tijdens het vooronderzoek de gegevensverzamelingen al in kaart zijn gebracht. Goed ontwerpen kost meer tijd dan goed inventariseren. Wanneer besloten wordt om te gaan automatiseren is het een heel goede beslissing om nog geen computer te kiezen, laat staan aan te schaffen. De moderne, in dit boek behandelde ontwerpmethoden maken het mogelijk gebruikers met een beeldscherm te laten werken zonder een computer aan te schaffen. Op die manier wordt tenminste het mogelijke gedaan om te voorkomen dat nieuwe computers al te klein blijken, als de helft van de toepassingen is gerealiseerd!
Konklusie: na een gedegen vooronderzoek begint het pas. De tijd die beschikbaar is voor het ontwerpen is evenredig met de kwaliteit van de te ontwikkelen informatiesystemen.
In de meeste bedrijven houdt men zich op strategisch niveau niet bezig met automatisering. Er worden GO/NO GO beslissingen genomen en budgetten getekend. Voor het strategisch managment is automatiseren een onderdeel van de administratie. Net zo min als men zich vroeger verdiepte in de aanschaf van een boekhoudmachine, houdt men zich nu bezig met die van een computer. Het strategisch management van veel bedrijven bestaat uit mensen die in de periode van de vijftiger tot de zeventiger jaren naar deze funkties zijn toegegroeid. Dat was een periode van groei, expansie, extrapolatie van stijgende lijnen en bijna automatisch sukses. Automatisering was een technisch proces dat geheel en al werd overgelaten aan de deskundigen.
Het tekenen van budgetten voor automatiseringsprojekten is voor veel direktieleden nog steeds weinig anders dan het tekenen van een blanko cheque. Het bedrag is wel ingevuld, maar er is meestal geen enkel inzicht in wat het bedrijf krijgt voor dat geld. In de periode van de tachtiger jaren staat het tekenen van blanko cheques echter wat meer onder druk dan vroeger. Automatiseren kan

Behalve aan beveiliging tegen misdrijven en rampen, dient men ook aandacht te besteden aan de beperking van de gevolgen. Brand of waterschade in een computerruimte betekent voor veel bedrijven een ramp die niet te overzien is. De computeruitwijk dient onderdeel te zijn van het bedrijfsrampenplan. In geval van brand of terreurdaden moet er meestal veel meer gebeuren dan alleen het uitwijken van het rekencentrum. Het zal niemand verbazen dat de projektaanpak om te komen tot een plan voor computeruitwijk (24) als twee druppels water lijkt op die van een gewoon projekt. Vereist is natuurlijk wel dat bij een uitwijkprojekt het uitwijkcentrum en het bedrijf dat wil uitwijken, met elkaar in de pas moeten lopen. Zo'n projektaanpak kent bij ontwerp, bouw en tests dan ook twee parallel lopende trajekten.
Een steeds terugkerend probleem bij uitwijkstudies is de bepaling van de benodigde kapaciteit van de computer, de schijven en het netwerk. Allereerst moet bepaald worden welke toepassingen in de uitwijksituaties moeten blijven draaien. Dat kan op zich al problemen opleveren, die eerder op politiek dan op technisch niveau liggen. De bepaling van de schijfkapaciteit is daarna relatief eenvoudig. De vaststelling van de benodigde machinekapaciteit voor batchtoepassingen is ook nog te doen. Het probleem ontstaat bij de interaktieve toepassingen. Vaak moet er een netwerk ontworpen worden dat gedurende de uitwijk de beeldschermen verbindt met de uitwijkcomputer. Nu is in het gemiddelde bedrjf het bestaande netwerk eerder ontstaan dan ontworpen en daarom zijn er geen gegevens bekend over hoeveelheden verkeer per toepassing per beeldscherm. Ook de benodigde machinekapaciteit voor interaktieve toepassingen is meestal slecht in kaart gebracht. In bedrijven met twee computercentra, die als elkaars uitwijkcentrum moeten dienen, heeft men vaak geen idee of een computer inderdaad alle kritische toepassingen aankan. De methoden in dit boek hebben alles te maken met interaktieve toepassingen en netwerkontwerp en dus ook met uitwijksituaties.

12.5 Strategische vergissingen

In veel bedrijven die beginnen met automatisering of die een minicomputer aanschaffen voor een bepaald bedrijfsonderdeel gaat de automatisering als volgt in z'n werk.
- De direktie vraagt advies over wel of niet automatiseren.
- Organisatie-adviseurs zijn een jaar of langer bezig met hun onderzoek.
- Het advies blijkt positief te zijn: automatiseren.
- Het strategisch management vraagt een kostenschatting en advies over de soort computer.
- De computer wordt besteld en moet zo snel mogelijk in bedrijf

liging van bijvoorbeeld niet geautomatiseerde bedrijfsonderdelen. Vandalen kunnen in het gemiddelde bedrijf op elke willekeurige afdeling in korte tijd een chaos aanrichten.
Naarmate de dreiging toeneemt, zullen we ons meer moeite en kosten willen getroosten om een afdoende beveiliging op te bouwen. Het beschrijven van de methoden valt buiten het kader van dit boek. In de vakliteratuur is er al genoeg over geschreven. Uiteindelijk gaat het om een goede risicoanalyse waarbij kosten van beveiliging tegen mogelijke verliezen worden afgewogen. Bij kosten gaat het dan niet alleen om de guldens, maar ook om het ongerief voor de gebruikers.
Er kunnen altijd maatregelen en technieken gevonden worden die afdoende beveiliging mogelijk maken. De kosten zullen zeker niet verwaarloosbaar zijn, maar wat betekent dat in een wereld, waarin de kosten van ieder automatiseringsprojekt achteraf een veelvoud blijken te zijn van de eerste schatting? Zoals altijd in de automatisering, speelt ook hier het probleem dat de deskundigen alles meteen vertalen naar de terminologie van hun vak. Vakliteratuur over beveiliging van computersystemen is geschreven voor en door automatiseringsdeskundigen. Wanneer echter het topmanagement greep wil krijgen op de beveiliging, dan moet kommunikatie mogelijk zijn in de gebruikerstaal. Met andere woorden: er moeten eisen gesteld worden door het gebruikersmanagement. Automatiseringsdeskundigen moeten die eisen vertalen naar oplossingen. De technische oplossingen moeten weer vertaald worden naar de kosten, de konsekwenties voor het bedrijf en de gebruikers. Gebruikers stellen dan eisen in hun taal en krijgen antwoorden in hun taal. Daarbij is het beoordelen van de resultaten van de risicoanalyse en het nemen van de uiteindelijke beslissing een zaak voor het taktisch en strategisch management. De konsekwenties voor de gebruikers achter het beeldscherm moeten echter een wezenlijke rol spelen in het beslissingsproces. Daar zal weer de kommunikatie tussen gebruikers en automatiseerders een rol spelen. Enerzijds vanwege de deskundigheid van de automatiseerders om alternatieven te bedenken, anderzijds in hun bereidheid de alternatieven te presenteren aan gebruikers, zodat die kunnen oordelen over de bruikbaarheid van diverse mogelijkheden. Natuurlijk zijn er technische beveiligingen mogelijk waar de gebruiker niets van merkt. Zo kan men bijvoorbeeld berichten volgens een bepaalde kode versluieren voor ze over een telefoonlijn worden verstuurd. Dat is een technische aangelegenheid waar de gebruiker achter het beeldscherm niets van merkt als het goed is. Dan gaat het alleen om het kostenaspekt. Samengevat kunnen we stellen dat computersystemen best afdoende beveiligd kunnen worden. De kosten zullen niet verwaarloosbaar zijn en daarom is het niet zo verstandig vooruit te lopen op de realiteit.

ling plaats naar de technische konsekwenties ten aanzien van de benodigde computer- en netwerkkapaciteit. Nu blijkt vaak dat computers al te klein zijn voordat de helft van de toepassingen gerealiseerd is. Als in de toekomst naast de gegevensverwerking ook allerlei kantoorautomatiseringsfunkties beslag leggen op de konfiguratie, wordt het kapaciteitsbeheer alleen maar belangrijker.

12.4 Beveiliging en uitwijk

Bij de diskussie over koppelingen tussen computers in verschillende vestigingen via telefoonlijnen komt bijna altijd de vraag op hoe veilig dat nu eigenlijk is. Men denkt daarbij aan het afluisteren van telefoonlijnen, het inbreken in computersystemen en dat soort zaken. Wanneer het gaat om de gebruikmaking van een openbaar datanet als Datanet 1, dan komt de vraag op, wat de kans is, dat de gegevens die daar met allerlei andere gegevens van andere bedrijven door de lucht vliegen, bij het verkeerde bedrijf terecht komen. Het zou toch niet best zijn wanneer de konkurrent onze inkoopprijzen te weten komt, doordat het Datanet een faktuur van onze leverancier bij de konkurrent aflevert. Misschien doet hij wel zaken met dezelfde leverancier! Het gaat dus om twee soorten beveiliging: tegen aktiviteiten van fraudeurs en tegen fouten van een netwerk.
Het is zinloos om te zoeken naar 100% beveiliging. Een dergelijke beveiliging kost onevenredig veel. Het gaat om een praktische beveiliging. Hoewel niemand van zijn huis een waterdicht beveiligde vesting zal maken, bestaan er wel degelijk een aantal praktische maatregelen die voldoende zijn om 90% van de inbrekers af te schrikken. Opvallend is daarbij dat 20 jaar geleden bijna niemand erover dacht en er nu allerlei mechanische en elektronische beveiligingen op de markt zijn. Niettemin zijn er nog steeds mensen die op plaatsen wonen waar de achterdeur dag en nacht onvergrendeld blijft. Kortom, de effektiviteit en het gebruik van beveiligingsmiddelen hangt af van de situatie en groeit met de dreiging. Computersystemen zijn slecht beveiligd. Zeker voor deskundigen in het automatiseringsvak. Misschien zijn ze theoretisch niet 100% te beveiligen, maar in de praktijk wel. Er zijn veel simpele maatregelen mogelijk, die nog niet worden toegepast. De reden daarvoor ligt voor de hand: hoe beter een systeem is beveiligd tegen ongewenste gebruikers, hoe onhandiger het wordt voor bevoegde gebruikers. Het onthouden van een wachtwoord is al niet leuk, maar iedere dag een ander wordt vervelend. Uiteindelijk gaat het dus om de afweging van kosten en overlast tegen de belangen die er mee gemoeid zijn. De toegang tot een drukkerij van bankpapier is anders geregeld dan die tot een schoensmeerfabriek. Verder kan het verhelderend werken eens te kijken naar de bevei-

gaan houden met automatisering. Het komt nu al regelmatig voor dat een direktie problemen heeft met de goedkeuring van budgetten omdat ze bijvoorbeeld geen idee heeft wat een network-interface-processor is, laat staan dat ze begrijpt waarom die moet worden aangeschaft. Bij kantoorautomatisering gaat het om veel grotere bedragen, terwijl de kosten/baten analyse alleen maar vager wordt.
Voor kantoorautomatisering is inzet op direktieniveau nodig. Naast de technische problemen is er een groot aantal organisatorische problemen op te lossen. Is de tekstverwerker een veredelde typemachine die wordt beheerd door het secretariaat of is het een computer die onder het rekencentrum valt? Is de schakelcentrale, die telefoongesprekken en gegevensverkeer regelt, een telefooncentrale of een computer van het rekencentrum? Wie beheert het netwerk dat tegelijk dient voor gegevensverkeer, spraak en tekst? Welke mensen zijn er nodig om kantoorautomatisering op te zetten? Automatiseerders, bedrijfskundigen of organisatie-adviseurs?
Een ander probleem is de aanpak. Om gebruikers te laten wennen aan het werken met beeldschermen kan men microcomputers beschikbaar stellen. Voor kantoorautomatisering zijn echter ook grote mainframes nodig en een netwerk om de kommunikatie mogelijk te maken. Naast het beeldscherm op de werkplek is dus een infrastruktuur nodig om een werkend geheel te krijgen. Dat betekent dat er beleidsuitspraken nodig zijn over de manier van werken van onderaf, bijvoorbeeld ten aanzien van microcomputers, en ten aanzien van de aanpak van bovenaf, bijvoorbeeld over funkties die mogelijk moeten zijn: archief, electronische post en dergelijke.
De aanpak is in grote lijnen zoals die bij gegevensverwerking zou moeten zijn: op basis van beleidsuitspraken worden uiteindelijk projekten vastgesteld en per projekt wordt er geanalyseerd, ontworpen, gebouwd, ingevoerd, beoordeeld en verbeterd.
Hoe het ook zal gaan met kantoorautomatisering, gegevensverwerking zal daar nog lange tijd een belangrijk deel van uitmaken.
Karakteristiek voor software voor kantoorautomatisering is het feit dat die nooit zelf ontwikkeld wordt, maar altijd gekocht. Niemand gaat zelf een tekstverwerker of een agendatoepassing ontwikkelen. Ook in de sfeer van gegevensverwerking zijn kant en klare pakketten op de markt. Veel bedrijven hebben echter processen die niet te vangen zijn in een standaardpakket. Als die al gekocht worden, moeten ze vaak aangepast of uitgebreid worden. Voor grote wijzigingen geldt in principe hetzelfde als voor nieuwbouw: er moet gewerkt worden volgens een projektaanpak waarin de kommunikatie met de gebruiker goed is geregeld.
De methoden die in dit boek worden behandeld betreffen het ontwerpen van gegevensverwerkende systemen. De inbreng van de gebruiker wordt daarin goed geregeld, maar er vindt ook een verta-

frastruktuur voor de informatie-uitwisseling. Die infrastruktuur moet worden afgeleid van het informatiebeleid, het informatiebeleid moet zijn afgeleid van het bedrijfsbeleid, en daarmee is de relatie met het strategisch management aangegeven. De uitspraak "meer micro`s, minder problemen", is net zo goed of slecht als het vastleggen van het bedrijfsbeleid met de uitspraak: "meer winst, minder kosten". De invoering van micro`s dient vergezeld te gaan van een ingevuld kader als boven is aangegeven.
Na verloop van tijd gaan de micro-eigenaars een paar dingen ontdekken.
- Alle funkties van de automatiseringsafdeling zijn nu samengeperst in een persoon: de micro-eigenaar.
- Een micro heet niet voor niets micro. Het is minder dan een mini. Dat mainframe heeft toch wel een aantal voordelen.
- Waarom kunnen wij niet kommuniceren met andere micro`s, mini`s en het mainframe?
- We zijn wel computer-eigenaars, maar toch geen automatiseerders. Met alle vragen moet je toch steeds weer naar de automatiseringsafdeling.

Het aantal problemen wordt zeker niet minder. Wanneer toch besloten wordt om micro`s in te voeren dan moet een aantal zaken zijn geregeld.
- De aanschaf moet centraal worden geregeld.
- De automatiseringsafdeling bepaalt welke merken mogen worden aangeschaft.
- De plaats van de automatiseringsafdeling in het geheel
  - Het beheer van alle micro`s en de software.
  - De ondersteuning van gebruikers voor- en achteraf.

12.3 Kantoorautomatisering

Kantoorautomatisering kan op allerlei manieren omschreven worden, afhankelijk van het aksent dat men wil leggen. Een van de definities zegt dat kantoorautomatisering betrekking heeft op de ondersteuning van de medewerker op de werkplek met behulp van elektronische hulpmiddelen, zodanig dat de taken efficiënt en effektief kunnen worden uitgevoerd.
Kantoorautomatisering betreft dus het hele gebeuren in een kantooromgeving. Bij de gegevensverwerking zoals we die al vele jaren kennen gaat het hoofdzakelijk om het administratieve gebeuren. Kantoorautomatisering is veel ingrijpender en tegelijk veelomvattender dan gegevensverwerking. Als we dus kantoorautomatisering op dezelfde manier aanpakken als gegevensverwerking staat ons nog wat te wachten!
Op strategisch niveau zal men zich toch uiteindelijk bezig moeten

zeker in de beginfase voortdurend op problemen. Dat betekent zeker bij grote aantallen micro-eigenaars een vaste kontaktpersoon binnen de EDP afdeling of een informatiecentrum. Daarnaast moet geregeld worden waarvoor de kommunikatiekanalen gebruikt kunnen worden. Als we kijken naar het soort problemen waarmee de nieuwe computereigenaars gekonfronteerd worden, dan is er in dertig jaar weinig veranderd. Nog steeds raakt hardware defekt zonder dat het gemeld wordt, nog steeds zijn er onleesbare floppies met belangrijke bestanden en nog steeds zijn computers "onvriendelijk" voor de first time user. Kortom, als we niet oppassen zijn binnen de kortste keren de kommunikatie-kanalen volledig verstopt en de gebruikers wanhopig.
In de tweede plaats dient de gebruiker opgeleid te worden voor hij kan besluiten om een micro te gaan gebruiken.Niet zozeer een opleiding in programmeren of een technische kursus over de werking van micro's. Het gaat om de vergelijking tussen het gebruik van een beeldscherm aan een grote computer en van een beeldscherm dat deel uitmaakt van een micro. Bij die vergelijking worden de mogelijkheden van micro's gerelativeerd. Dan kan duidelijk gemaakt worden waarom er zoveel specialisten rond lopen op de automatiseringsafdeling en wordt de gebruiker zich bewust van het feit dat hij dus eigenlijk informatie-analist, systeemontwerper, programmeur, beheerder en operator tegelijk moet zijn. Op dat moment worden ook de mogelijkheden en beperkingen duidelijk van de veel geprezen, maar weinig betekenende koppeling tussen de microen de grote computer. Wanneer daarnaast enig inzicht is gegeven in wat er komt kijken bij het kiezen van een programmeertaal, de keuze van een pakket en het kiezen van hardware, krijgt de gebruiker alsnog de vrijheid om te kiezen tussen gebruik maken van de diensten van de centrale automatiseringsafdeling of voor zichzelf beginnen.
In de derde plaats dient voorzien te worden hoe de micro's, op de diverse lokaties, geintegreerd kunnen worden tot een geheel. Nogmaals: micronisering kan een vlucht zijn. Wanneer de centrale EDP de gebruikers niet kan bevredigen, geef de klagers dan hun eigen computer. Dan kunnen ze hun eigen prioriteiten stellen en hun eigen problemen oplossen. Het klinkt als een duurzame oplossing, maar het is een zeer tijdelijke. Na korte tijd gaan de eigenaars de beperkingen van hun rekentuig zien. Zeker de gebruikers die gewend waren aan de gigantische gegevensbestanden op mainframes. De responsetijden op mainframes zijn vaak niet ideaal, maar veel gebruikers realiseren zich niet dat ze werken met toepassingen die zo komplex zijn, dat ze nooit met een micro kunnen worden gerealiseerd. Vroeg of laat ontstaat de vraag naar samenwerking tussen de micro en de grote systemen binnen het bedrijf of daarbuiten. In feite is het de vraag naar een goede in-

Kortom, wanneer voor decentralisatie wordt gekozen vanwege negatieve kritiek op de centrale automatisering, dan zal het middel erger lijken dan de kwaal. Even afgezien van de argumentatie blijkt er wel een goede weg te zijn naar decentralisatie:
- gemethodiseerde analyses
- gemethodiseerd netwerkontwerp.

12.2 Microcomputers

Decentralisatie kan een vlucht zijn. Wanneer een direktie de centrale automatisering niet meer beheerbaar acht en de klachten van de gebruikers redelijk lijken, dan moet er iets gebeuren. Van een topmanagement worden beslissingen verwacht. De automatisering wordt dan meestal gedeeltelijk naar de lokale vestigingen verschoven. Er worden mini`s aangeschaft. In het begin van de tachtiger jaren tekent zich een nieuwe ontwikkeling af: de microcomputer wordt aangeprezen als de nieuwe oplossing zijn voor de professionele automatisering. Hobbycomputers of huiscomputers voor de prive-liefhebbers, professionele computers voor de zakenmensen. In plaats van een beeldscherm heeft nu iedereen een komplete computer op zijn bureau. Gebruikers zijn eigenaars geworden. De centrale automatiseringsafdeling is nu helemaal overbodig geworden, want iedereen maakt zijn eigen programma`s, selekteert zijn eigen pakketten en lost zijn eigen problemen op. En dat alles natuurlijk veel sneller en effektiever dan die grote, logge automatiseringsafdeling.
Zo kan het een uitstekende oplossing lijken om in het hele bedrijf over te schakelen op micro-computers. Evenals bij de minicomputers in de zeventiger jaren, een ingrijpende beslissing, maar wanneer ze genomen wordt om eindelijk af te zijn van de klaagzangen van de gebruikers, dan is ook dit een vlucht. Dan zal ook nu, het middel erger blijken dan de kwaal. Elke ongenuanceerde, generale beslissing levert in de praktijk een aantal problemen op. Het lijkt strategisch om alleen in grote lijnen te denken, maar veel strategen zijn gestruikeld over enkele praktische probleempjes. Het klinkt strategisch om "bedrijfsbreed" over te gaan tot de aanschaf van micro`s. Overschakeling op micro`s zonder een kader aan te geven is hetzelfde als geen uitspraak doen en gebruikers zelf te laten uitmaken of ze een micro aanschaffen. Een kader aangeven is meer dan een bepaald fabrikaat voorschrijven, hoewel dat voor veel bedrijven al heilzaam geweest zou zijn. Er is een aantal zaken dat binnen dat kader moeten worden aangegeven.
In de eerste plaats de relatie tussen de bestaande automatiseringsafdeling en de nieuwe computereigenaars. De kommunikatiekanalen dienen te worden vastgesteld. De nieuwe eigenaars stuiten

Deze vragen worden nog dringender wanneer het gaat om internationale koppelingen. Als nationale vestigingen zelf hardware mogen kiezen en zelfstandig software ontwikkelen of laten ontwikkelen is de ramp niet te overzien. Afgezien van het feit dat steeds weer tegen hoge kosten het wiel wordt uigevonden, zullen de netwerkkosten onvoorspelbaar en dus hoog worden. En ook van decentralisatie moet gezegd worden: er is geen weg terug.
Als een centrale automatiseringsafdeling al niet bestuurbaar is, hoe zou het dan gaan met een veelvoud van kleine automatiseringsafdelingen?
De wet van behoud van ellende zal een eufemisme blijken.
De som van alle "lokale ellende" is veel groter dan de "centrale ellende". Er is geen weg terug, wel een goede weg heen:
- Een konkreet op schrift gesteld bedrijfsbeleid. Dat is dus meer dan de uitspraak: minder kosten en meer winst.
- Een konkreet op schrift gesteld informatiebeleid dat afgeleid is uit het bedrijfsbeleid.
- Een konkreet op schrift gesteld automatiseringsbeleid, afgeleid van het informatiebeleid.
- Het verplicht stellen van het toepassen van uitgekristalliseerde methoden ten aanzien van
  - inspraak van gebruikers,
  - analyse van de hoeveelheid verkeer binnen het netwerk,
  - integratie van programma-, gegevens- en netwerkontwerp.

In het automatiseringsbeleid moet wat het netwerk betreft minstens het volgende zijn vastgelegd.
- De verhouding tussen centrale en de-centrale automatiseringsafdelingen. Zowel qua organisatie als qua kommunikatie.
- Bevoegdheden en verantwoordelijkheden van het lokale management.

Soms is binnen een bedrijf reeds gekozen voor een systeemontwikkelingsmethode. Dan moet kunnen worden aangegeven hoe daarin het netwerkontwerp gerealiseerd wordt. In de meeste methoden is alle plaats ingeruimd voor programma-ontwerp en database-ontwerp. Het netwerk blijkt bijna nooit ontworpen te zijn. Er worden lijnen gehuurd en er wordt transmissie-apparatuur aangeschaft. De leveranciers van deze apparatuur zijn meestal zeer deskundig op het gebied van datakommunikatie. Ze worden er echter niet voor betaald om de toepassingen qua hoeveelheden verkeer te analyseren. Daarom gaan ze uit van wat vuistregels en ervaringscijfers en daarin ligt dan tevens de oorzaak van een falend netwerk. De automatiseringsafdeling vindt het netwerk een zaak van technici en de technici verwachten kwantitatieve gegevens van de automatiseringsafdeling. In systeemontwikkelingsmethoden is daarin echter niet voorzien. Transaktie analyse, zoals dat in dit boek zal worden beschreven, kan in dit opzicht een goede bijdrage leveren.

en micro's in kantoren. Meestal wordt het centrale mainframe niet afgeschaft, dus vroeg of laat moet er dan toch een netwerk ontworpen worden om de mini's met het mainframe te verbinden. Al zou het alleen maar zijn omdat op het mainframe de centrale boekhouding draait. De redenen voor die decentralisatie kunnen uiteenlopen. Er kunnen goede, funktionele redenen voor bestaan, bijvoorbeeld bijzondere toepassingen die een heel ander soort computer of heel andere randapparatuur vergen.
Het wordt anders wanneer gebruikers gaan aandringen op decentralisatie, zeker als daar alleen argumenten voor worden gebruikt die negatief zijn ten aanzien van de centrale automatiseringsafdeling. Dan zijn de gebruikers kennelijk ontevreden over het funktioneren van die afdeling. Of dan het "bezit" van een eigen minicomputer een oplossing is, valt te betwijfelen.
Erger wordt het wanneer het topmanagement besluit de gebruikers hun eigen mini of micro te geven om een eind te maken aan hun geklaag. In een tijd van bezuinigingen en goedkoper wordende hardware is het natuurlijk een prima gedachte de grote automatiseringsafdeling wat af te slanken en de gebruikers hun eigen boontjes te laten doppen met hun eigen computer. Wanneer dat de basis wordt voor de decentralisatie is er sprake van een vlucht. Of de gedecentraliseerde toekomst er beter uitziet dan het centrale heden is de vraag. Er ontstaan andere, minstens zo vervelende problemen.
Eigenlijk is er eerder sprake van een vlucht dan van een beleid. In de praktijk blijkt dat er altijd een vorm van kommunikatie tussen de computers van de vestigingen en het centrale mainframe nodig is. Hoe onafhankelijk de vestigingen ook zijn, er is altijd een vorm van centraal management en voor management is informatie onontbeerlijk. Dus worden er netwerken ontworpen, en blijken computers niet met elkaar te kunnen kommuniceren. De totale kosten kunnen gemakkelijk veel hoger zijn dan die van de centrale situatie. In de zeventiger jaren werden decentraal mini's geplaatst. De lokale vestigingen kregen daarmee een stuk zelfstandigheid in de automatisering. Deze vorm van decentralisatie wordt nog steeds toegepast en het kan ook best een prima oplossing zijn. Waar het om gaat is hoe de beslissing tot stand is gekomen en of de voor- en nadelen voldoende zijn overzien. Van groot belang daarbij is de vakkennis rond het ontwerpen van een netwerk. Het netwerk moet immers van de diversiteit aan computers een konsistent en bruikbaar geheel maken. Essentieel is de plaats van de centrale automatiseringsafdeling. Het gaat daarbij om vragen als

- Welke kommunikatie kanalen zijn nodig?
- Wie beheert de hardware en de software?
- Wie beheert het netwerk?
- Hoe worden de netwerkkosten doorbelast?

Hoofdstuk 12 Enkele aspekten van de automatisering

12.1 Decentralisatie

De automatisering begon in veel bedrijven met de aanschaf van een computer. Naarmate meer bedrijfsonderdelen werden geautomatiseerd groeide en groeide de computer. Met woorden als mainframe en hostcomputer bedoeld men nog steeds de centrale computer. Met de groei van de computer groeide meestal ook de automatiseringsafdeling. Ieder jaar waren grotere budgetten nodig voor de automatisering. Wat er niet groeide was de tevredenheid van de gebruikers. Zij begonnen te klagen over de kwaliteit van de produkten van de automatiseringsafdeling en over de levertijden. Daarnaast bleek vaak dat de automatiseerders onmachtig waren de gebruikers op een lijn te krijgen. Zeker bij grotere bedrijven met vele vestigingsplaatsen lag dat vaak niet aan de automatiseerders!
Een direktie ziet dus twee zaken: een erg dure centrale automatiseringsafdeling en ontevreden gebruikers. Dan lijkt decentralisatie de oplossing. Ontmanteling van de centrale automatiseringsafdeling, en lokale managers mogen hun eigen automatisering opzetten. Dit soort beslissingen grijpt zeer diep in in de infrastruktuur voor de informatie voorziening van het bedrijf.
Er kunnen heel goede redenen zijn om van het centrale computercentrum af te stappen en over te gaan tot het plaatsen van mini's

Op dat moment worden de sociale aspekten van de automatisering voor hem heel konkreet. Hij kan meepraten over de gevolgen voor zijn werk, voor zijn taak/funktie-omschrijving, voor de manier van werken binnen de afdeling, enzovoort. Bij dat alles is hij geen automatiseerder geworden, hoeft hij niet te weten hoe een computer werkt en geen kursussen burgerinformatica te volgen. Er zijn kursussen als (15) waardoor hij in een paar dagen wordt opgeleid tot mede-ontwerper van interaktieve systemen. Alles wat in kursussen wordt verteld over techniek moet behandeld worden vanuit het gebruik van computers en in dienst staan van de kommunikatie met automatiseerders. Als daarnaast gekozen wordt voor begeleiding van gebruikers door "softe" opleidingsinstituten dan is de keuze in handen van het taktisch management, maar het is van belang dat men zich op strategisch niveau realiseert dat er vele soorten instituten zijn en dat ze opereren vanuit een bepaalde visie op de maatschappij. Op de een of andere manier klinkt die visie door in de trainingssessies. Het is van belang zich ervan te vergewissen of die visie aansluit bij de visie van de direktie op het bedrijfsgebeuren en op de automatisering.
Samengevat kunnen we vaststellen dat er, terugziend op twintig jaar automatisering, banen verdwijnen, verschijnen en verschuiven. Zolang de verhouding tussen die drie zaken niet is vastgesteld heeft het weinig zin alleen over de eerste te praten. Met name het verschuiven, het veranderen van de inhoud van een baan, moet voor de funktionaris zelf volkomen duidelijk zijn, voordat met de bouw van een systeem wordt begonnen. Daarbij gaat het om het kwalitatieve: wat moet er gedaan worden en om het kwantitatieve: Hoe lang duurt het? Hoe wordt de dagindeling en hoe zit dat in pieksituaties?

Het ontwerpen van interaktieve toepassingen
en computernetwerken.

Handleiding voor strategisch en taktisch
management, gebruikers en automatiserings-
specialisten.

grijpbare sociale aspekten. Het meest konkrete probleem ontstaat wanneer een werknemer hoort dat hij in zijn werk met een beeldscherm te maken krijgt. Een deel van het papier- of handwerk vervalt en wordt overgenomen door de computer. Over die situatie gaat het bij dit onderwerp.
Algemene verhalen over chip en werkeloosheid zijn al vaak verteld en geschreven, maar opgelost hebben ze niets. Wat is er precies aan de hand met de werknemer? Hij heeft gehoord dat de automatisering ook hem heeft bereikt. Hij wordt geinterviewd door informatie-analisten. Wat ze vragen is misschien nog duidelijk, maar waarom ze het vragen en wat er met de antwoorden gebeurt, ontgaat hem.
Vragen over zijn toekomstige werk worden in een onbegrijpelijk jargon beantwoord. In het gunstigste geval wordt hem dan ook nog gevraagd zoveel mogelijk mee te denken, mee te praten en mee te doen. Hij ontvangt soms pakken dokumentatie waarvan hij alleen aan de titel kan zien dat het zijn werk betreft, de inhoud is voor hem onleesbaar. Hij begrijpt dat er inmiddels een legertje specialisten in de weer is om zijn werk "op de computer te zetten". Hij heeft ook gehoord dat het volgens de planning nog een jaar duurt. Dat planningen gewoonlijk uitlopen weet hij allang. Welk deel van zijn werk er nu precies geautomatiseerd wordt en hoe dat dan zal gaan kan niemand hem duidelijk maken.
Hij wordt steeds verwezen naar pakken papier met een funktioneel ontwerp, beeldschermlay-outs, bestandsbeschrijvingen enzovoort. Zo leeft deze werknemer lange tijd tussen hoop en vrees. Zal hij zo`n beeldscherm kunnen bedienen? Wat gebeurt er als hij iets fout doet? Wat komt er eigenlijk op zo`n scherm te staan? Dat is het konkrete probleem voor de werknemers en hun konkrete werksituatie. Het vervelende daarbij is dat ze het ergste nog niet eens weten. Ze weten niet dat wat er voor hen ontworpen en gebouwd wordt, onveranderlijk is, als het gereed is en gedemonstreerd wordt.
Afgezien van wat details is de manier van werken voor vele jaren vastgelegd. Met de methoden die in dit boek verder worden uitgewerkt, wordt het mogelijk de gebruiker in de ontwerpfase, wanneer er nog niets gebouwd is, de ervaring te geven alsof hij reeds achter het beeldscherm zit. Tot zijn opluchting merkt hij dat het allemaal niet zo moeilijk is en dat er niet zoveel fout kan gaan. Na enige tijd met beeldscherm te hebben gewerkt, merkt hij zelfs dat alternatieven of andere voorstellen binnen een uur ook weer werken en ervaren kunnen worden. Hij merkt dat er andere oplossingen zijn en dat de informatie-analist er geen probleem van maakt om die ook even op het scherm te zetten. In korte tijd begint hij de mogelijkheden van computers te begrijpen en voelt hij zich mede-ontwerper.

slechte responsetijden.
Dat rapport kan zo naar de direktie. Dan wordt het stil.
Wat geldt voor responsetijden, geldt voor extra kosten, meer tijd, meer apparatuur, onhandige bediening, enzovoort.
Praktisch alle problemen vinden hun oorzaak in kommunikatiestoornissen tussen gebruikers en automatiseerders tijdens het ontwerpproces.
Als er op papier staat dat gebruikers moeten worden ingeschakeld bij het ontwerpproces, zonder de methoden worden aan gegeven, hangt de kommunikatie af van de persoonlijke inzichten van iedere automatiseerder. Daarmee staat feitelijk de kommunikatie op losse schroeven. Het proces is ook niet te managen, want de gebruiker is altijd ingeschakeld. Al zal het maar zijn omdat hij verslagen van vergaderingen heeft ontvangen.
Veel automatiseerders voelen zich beperkt in hun kreativiteit wanneer ze volgens methoden moeten werken. In feite vragen ze om de vrijheid voor gebruikers een surprise te maken, waarvan de voorbereiding goud kost. In plaats van hun kreativiteit te gebruiken in dienst van de kommunikatie met gebruikers, leggen ze zich toe op het verrassingseffekt. Tot aan de oplevering blijven ze denken dat gebruikers verrast zullen opkijken als ze met dat geweldige systeem mogen werken! De anti-climax werkt frustrerend. Hoewel ze het nooit toe zullen geven, is het werken volgens een vaste methode veel minder erg. De kommunikatie met de gebruiker is zo wezënlijk in het gebeuren, dat het resultaat ervan getoetst moet zijn tijdens het logisch ontwerp. Tijdens die fase moeten gebruikers precies weten wat er bij de oplevering gaat gebeuren. Gebruikers moeten verplicht worden op te treden als mede-ontwerpers. Het moet ze onmogelijk gemaakt worden zich aan het ontwerpproces te onttrekken. De kommunikatie moet verlopen volgens een gekozen, te managen methode. Daarover is geen diskussie mogelijk. dus toch de zweep er over!

11.6 Sociale aspekten van de automatisering

In een boek over netwerkontwerp wordt nu niet direkt een verhandeling verwacht over de sociale aspekten van automatisering. We zullen dan ook direkt het kader aangeven en de reden ertoe noemen. We beperken ons tot de beeldschermsituatie. Het gaat dus om werkplekken die uitgerust zullen worden met een beeldschermen en eventueel een printer. De reden voor dit onderwerp is het feit dat de methoden die in dit boek behandeld worden een neveneffekt hebben dat van belang is voor de sociale aspekten van de automatisering.
Het gaat dan ook niet om een filosofische, alles overziende behandeling van het onderwerp. Neen, het gaat om enkele konkrete,

Als de gebruiker wil weten wat er gebeurt als het aantal orders toeneemt tot 1200, heeft hij binnen een uur antwoord. Zo kunnen allerlei situaties worden doorgerekend tijdens het logisch ontwerp !
In het deel voor het taktisch gebruikersmanagement wordt een en ander verder uitgewerkt, terwijl in het deel voor de gebruikers precies is aangegeven hoe dialoogsimulatie, in samenwerking met informatie-analisten moet worden uitgevoerd. Transaktie analyse voeren automatiseerders uit.

11.5 De zweep erover?

Het is natuurlijk niet meer van deze tijd om nog te spreken over het gebruik van een zweep bij het managen van mensen. Laten we daarom de zweep voorlopig in de hoek zetten.
Via de pers worden we regelmatig op de hoogte gehouden van het wel en wee van de automatisering bij de overheid. De algemene rekenkamer rapporteert over een groot aantal projekten dat niet is afgerond, niet tot invoering van het systeem heeft geleid, onbruikbaar bleek of is blijven steken in misverstanden. Voor veel bedrijven is het maar goed dat er geen rekenkamer is die hun projekten evalueert en in de openbaarheid brengt. Het gaat bij automatisering vaak om enorme investeringen, terwijl bekend is dat het gemakkelijk fout kan gaan en dat de afloop in ieder geval onzeker is. De kosten van de automatisering zullen alleen nog maar stijgen. Niet omdat de hardware niet goedkoper wordt, maar omdat we steeds meer funkties willen realiseren die meer hardware en inzet van specialisten vragen.
Nu is het eenvoudig om bij alle min of meer mislukte projekten naar de automatiseerders te wijzen. Als bij huwelijksproblemen geldt, dat beide partners evenveel schuld treft, dan ligt ook in de automatisering de helft van de schuld bij de gebruikers.
Laten we als voorbeeld de responsetijd nemen. Dat is de tijd die men moet wachten op een reaktie van de computer. Zelfs de meest ontechnische direkteur kan beoordelen of een data-entry-typiste vlot kan werken met een systeem of dat de responsetijden inderdaad onhandig lang zijn. Stel dat hij aan het hoofd automatisering vraagt waarom de responsetijden zo lang zijn, die op zijn beurt zijn specialisten om een rapport over de oorzaken vraagt. Het is niet uitzonderlijk als hij van iedere specialist een rapport krijgt over een deel van de oorzaken. Het is de vraag wat hij met vijf rapporten moet. Het slimste rapport komt echter van een projektleider: Het ontwerp was zo goed, dat de responsetijden flitsend hadden kunnen zijn. Helaas kwamen de gebruikers op allerlei momenten nog met eisen, die van het oorspronkelijke koncept niets overlieten. De aanpassingen zijn de oorzaak van de

- De gestelde eisen moeten direkt kunnen worden omgerekend naar gevolgen voor het werk en de werktijden van de gebruikers.
- De methoden moeten aansluiten bij de methoden die automatiseerders gebruiken, want zij moeten de technische konsekwenties van de eisen kunnen berekenen.

Het gaat om twee methoden: dialoogsimulatie en Transaktie analyse.
Bij dialoogsimulatie wordt de dialoog met de computer, dus het werken met een beeldscherm gesimuleerd. De dialoogsimulator is een softwarepakket dat draait op een P.C. en het beeldscherm van de P.C. laat funktioneren zoals later een willekeurige toepassing zal funktioneren. Daarbij hoeft niets geprogrammeerd te worden. De schermlay-out wordt vastgelegd en er kan gewerkt worden. In het algemeen zal een informatie-analist de simulatie voorbereiden en de gebruikers zullen de simulatie uitvoeren bijvoorbeeld met gebruikmaking van hun dokumenten. Telefonische transakties kunnen gesimuleerd worden via een intern telefoontoestel. De P.C. is draagbaar en op iedere werkplek, op iedere vestiging kan gesimuleerd worden.
Dialoogsimulatie is een vorm van prototyping. Bij prototyping moet echter geprogrammeerd worden, er moeten bestanden aanwezig zijn en er moet een computer aanwezig zijn. Dat betekent dat het allemaal wat langer duurt en dat wijzigingen aanpassingen in de software betekenen. Automatiseerders werken vaak liever met prototyping omdat ze dan ook programmafunkties en gegevens hebben getest. Een gebruiker is alleen geinteresseerd in de dialoog: wat moet hij intypen en wat zet de computer op het scherm? Zelfs al stelt hij belang in de verwerking door de computer, hij realiseert zich die verwerking niet konstant als hij met het beeldscherm werkt. Dialoogsimulatie brengt het werken met beeldschermen kwalitatief tot leven.
Dialoogsimulatie wordt gevolgd door Transaktie analyse: de resultaten van dialoogsimulatie vormen de start voor Transaktie analyse.
Transaktie analyse is in principe een rekenproces, dat de dialoog met de computer en alle menselijke handelingen kwantitatief in kaart brengt. Uit dat rekenproces komen twee soorten gegevens tevoorschijn: ergonomische en technische.
De ergonomische resultaten geven informatie over de gebruikerskant van het beeldscherm, de technische resultaten over de grootte van het computersysteem, de kapaciteit van het netwerk enzovoort. Stel dat een gebruiker zegt dat hij op zijn afdeling 1000 orders per dag verwerkt. Met dialoogsimulatie wordt het invoeren van orders ontworpen en uitgetest door alle betrokken gebruikers. Transaktie analyse levert resultaten op als aantal beeldschermen, aantal beeldschermuren, kosten van het netwerk en dergelijke.

der. Ze moeten het bedrijf en al zijn problemen in kaart gaan brengen. Daartoe wordt midden in het bedrijf een hoge toren gebouwd, van ivoor. Vanuit die toren overzien ze het bedrijf, de mensen, de processen, de gegevens. Dat brengen ze allemaal in kaart op dokumenten vol vierkantjes, ruitjes, rolletjes en bolletjes. Wanneer de direktie aan de mensen in de toren vraagt wanneer de computer nu gaat werken dan komt er een lang antwoord dat erop neerkomt dat de toren nog niet hoog genoeg is en dat er nog niet genoeg mensen werken.
Zo langzamerhand begint de rest van het bedrijf, vroeger gewoon het bedrijf, zich af te vragen wat dat allemaal kost en wanneer er eindelijk eens iets gebeurt. Na verloop van enkele jaren is er natuurlijk wel iets bereikt: de salarisadministratie, de urenregistratie, de voorraad en de boekhouding zijn geautomatiseerd. De planningen zijn uitgelopen, de kosten waren hoger, de computer bleek te klein en de gebruikers zijn niet gelukkig met het hele gebeuren maar verder loopt het prima.
De direktie heeft er een gigantisch probleem bij gekregen. Er zijn tonnen geinvesteerd in iets wat maar niet klaar komt. Wat wel gereed is spreekt de gebruikers niet aan en tot overmaat van ramp begint het iedereen duidelijk te worden dat er geen weg terug is.
Wij gaan nog even terug naar het begin. Om bepaalde redenen besluit de direktie een computer aan te schaffen. In ieder geval is het iedereen duidelijk dat het een gereedschap moet worden voor het bedrijf en dus voor alle werknemers. Dan is maar een ding van belang: de gebruikers moeten tevreden, eigenlijk enthousiast zijn over de nieuwe manier van werken! De kern van goed automatiseren is dus: vanaf de eerste dag die aan een projekt besteed wordt, zeker weten dat er iets ontwikkeld wordt, wat de gebruikers wensen. Daarom is van groot belang dat het topmanagement zich van tevoren laat demonstreren volgens welke methoden automatiseerders omgaan met gebruikers.
Methoden die het mogelijk maken van het ontwerpen een iteratief proces te maken moeten aan een aantal eisen voldoen.

- De gebruiker moet de toekomstige situatie op zijn werkplek kunnen ervaren. Daarbij gaat het om alle menselijke handelingen en de dialoog met de computer. De verwerking door de computer wordt nog buiten beschouwing gelaten.
- De methoden moeten onafhankelijk zijn van computers. Zelfs als er helemaal nog geen computer is aangeschaft, moet de gebruiker met het beeldscherm kunnen werken zoals hij dat later eventueel zal doen.
- De gebruiker moet gaan begrijpen wat realistische eisen zijn zowel ten aanzien van de bediening als ten aanzien van responsetijden.

voorstellen per werkplek en als dan de konsekwenties van de automatisering voorgerekend zouden worden, zouden ze eisen aan het systeem kunnen stellen, waar ze anders mee aankomen als het te laat is. Anders gezegd, evaluatie tijdens het ontwerp. Zie Fig. 11.1.
In het volgende hoofdstuk worden methoden besproken die het mogelijk maken gebruikers tijdens het logisch ontwerp met het beeldscherm te laten werken alsof het systeem al gebouwd was! Bovendien kan op dat moment ook worden berekend hoeveel beeldschermen per afdeling nodig zijn, hoeveel uren het verwerken van transakties kost enzovoort. In feite komt het er op neer dat het iteratieve proces zich niet afspeelt over een projekt heen, maar binnen de ontwerpfase.
Het is opvallend hoe vaak direktieleden enerzijds praten en denken over automatisering als over iets dat aangeschaft wordt en alleen nog even gemaakt moet worden en anderzijds vaak wel iets hebben gehoord over de problemen rond het werken met beeldschermen. Ze schaffen iets aan, terwijl ze weten dat het precies moet passen in het bedrijfsgebeuren. Ze denken over maatwerk, in termen van konfektiewerk. Het kernpunt van automatiseren ligt in het ontwerpproces. Alles wat daar fout gaat, alles wat daar vaag en onnauwkeurig blijft, heeft enorme gevolgen voor het uiteindelijke resultaat.

11.4 De ivoren toren van Babel

Laten we eens kijken hoe in grote lijnen de automatisering in veel bedrijven begonnen is.
Op een gegeven moment wordt de direktie zich bewust van het feit dat er nog geen computer is. Het kan zijn dat een van de direktieleden een buurman heeft die eveneens direkteur is en hem duidelijk heeft gemaakt hoe snel je in deze maatschappij achter loopt. In zijn bedrijf heeft men nu de knoop doorgehakt: er komt een computer. Zo simpel is dat, want daarna gaat alles immers sneller, beter en met minder kosten. Het kan ook zijn dat de direktie ziet dat de konkurrenten overgaan op de computer. Wat ze er precies mee doen is niet onderzocht maar stel je voor dat ze morgen met lagere prijzen op de markt komen of meer service bieden!
Zo zijn er nog meer redenen te noemen om een computer aan te schaffen.
Bij een computer horen specialisten zoals informatie-analisten, systeemontwerpers, programmeurs, operators, systeembeheerders. kortom het wordt een bedrijf in het bedrijf. En nog een bijzonder bedrijf ook. Bijzondere mensen met bijzondere salarisschalen, een eigen taal en een eigen schrift. Ook de huisvesting wordt bijzon-

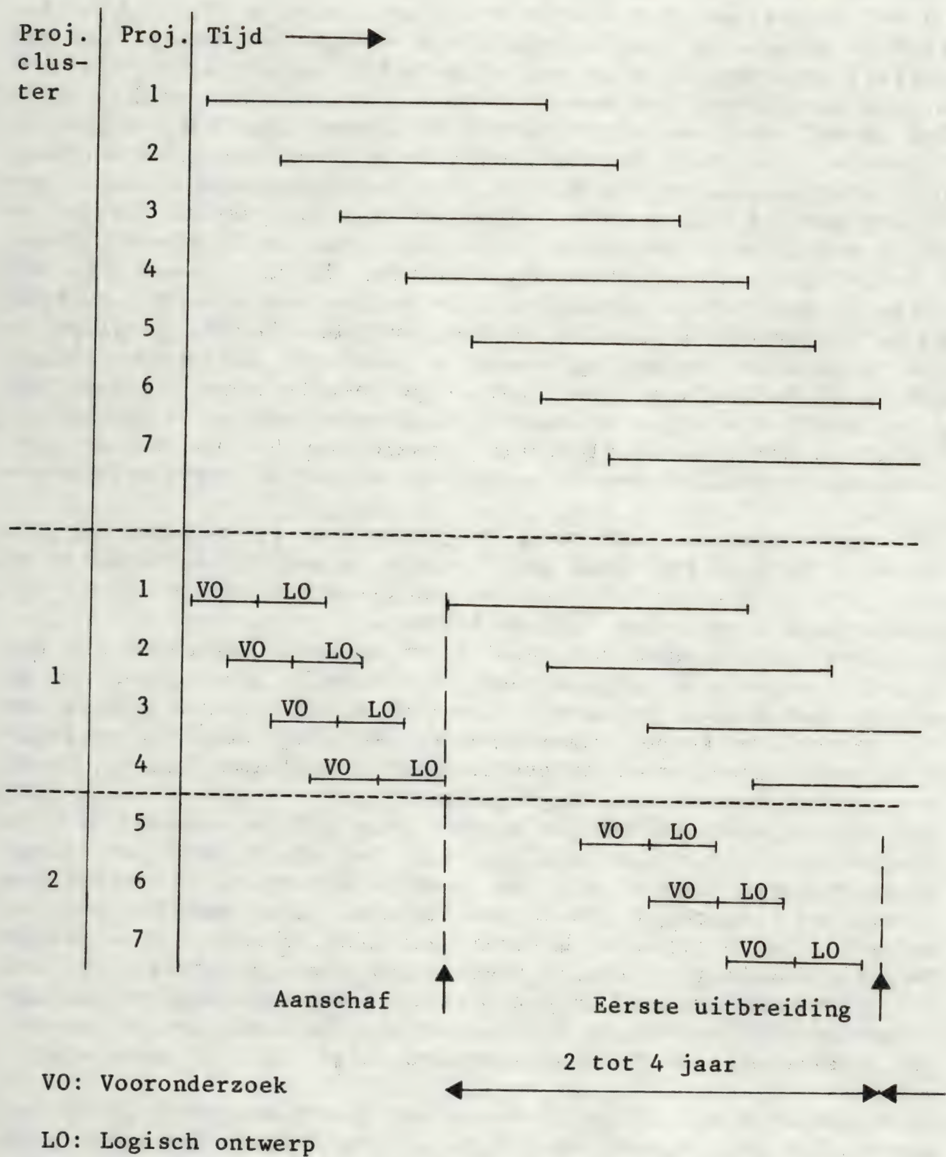


Fig. 31.1 Projektafhandeling zonder en met transaktie-ontwerp.

formance te onderhandelen.
Op het moment dat er besloten wordt een computer aan te schaffen zijn er meestal een aantal projekten vastgesteld die in een bepaalde volgorde zullen worden uitgevoerd. Vaak zijn daarbij termijnen aangegeven. Een aantal projekten zal zo snel mogelijk worden gerealiseerd om aan de meest dringende behoefte van de gebruikers te voldoen, andere projekten zijn wel voorzien maar nog niet precies gepland. In Fig. 31.1 is een ongunstige situatie getekend waarbij een zevental projekten zijn vastgesteld die de een na de ander gerealiseerd zullen worden, met een doorlooptijd van 4 tot 6 jaar. Om met transaktie-ontwerp de performance van het systeem goed te beheren zullen prioriteiten moeten worden gesteld en de projekten in clusters worden verdeeld. In veel gevallen is een dergelijke groepering bewust of onbewust al gemaakt, zeker door gebruikers die met het laatste projekt te maken krijgen. Per groep wordt het logische ontwerp uitgevoerd voor alle projekten. Per groep wordt de konfiguratie bepaald aan de hand van de cijfers uit transaktie-ontwerp en in overleg met de computerleveranciers.
Het realiseren van een groep projekten tot en met het logisch ontwerp, is mogelijk omdat door transaktie-ontwerp de akseptatietest, die anders aan het eind van een projekt uitgevoerd, nu plaatsvindt tijdens het logisch ontwerp.
Bij P5-projekten gaat het meestal om bestaande systemen die gekoppeld worden. De toepassingen die gebruik gaan maken van de koppelingen moeten geanalyseerd worden met Transaktie analyse om het netwerk te kunnen dimensioneren. Als de nieuwe toepassingen problemen zouden kunnen opleveren voor de systeembelasting, wordt het wat dat betreft een P3-projekt.
Bij P6-projekten worden de meeste vergissingen gemaakt wat de doorlooptijd van het projekt betreft. Nog steeds wordt er op managementniveau gedacht aan een simpele konversie: wat eerst geprint werd verschijnt nu op het scherm. Alle funkties blijven hetzelfde, alle gegevens blijven hetzelfde, alleen kan nu de informatie interaktief worden opgevraagd. Deze uitspraken bewijzen een gebrek aan inzicht in het ontwerpen van interaktieve toepassingen en een ontkenning van negatieve ervaringen op dit gebied. Dialoogsimulatie en Transaktie analyse zijn ook hier de sleutelwoorden.
P7-projekten komen meestal neer op het uitvoeren van aktiviteiten die tijdens het ontwerp niet zijn gedaan. Tijdens het ontwerp is iedereen druk doende met zijn aandeel in het projekt en niemand is verantwoordelijk voor de performance en de responsetijden. De database-ontwerpers kijken naar de programma-ontwerpers, de programma-ontwerpers naar de systeemspecialisten maar van netwerken

en datakommunikatie weet men in deze drie vakgebieden meestal weinig. Wat de knelpunten in de responsetijden zouden kunnen zijn is nog wel bekend, maar niemand weet wat er in een gegeven situatie aan de hand is. Transaktie analyse brengt de interaktieve toepassingen in kaart. Daarbij ontstaan gegevens over de vertraging ten gevolge van het netwerk, over de verwerking door het computersysteem en over de belasting die de transakties opleveren voor het systeem. Op die manier ontstaat inzicht in de oorzaak van lange responsetijden.
Voor de evaluatie van de ergonomie zijn externe adviseurs meestal de beste scheidsrechters tussen gebruikers en ontwerpers. Het is vaak verrassend te zien hoe gebruikers fouten in de bediening aanwijzen die ontwerpers met enige basiskennis van dialoogontwerp toch niet meer hadden mogen maken. Daarbij moet wel bedacht worden dat de klachten van de gebruikers vaak maar een fraktie van alle aspekten van de bediening betreffen. Bij een evaluatie horen ook de goede kanten van een ontwerp naar voren te komen. Ook al worden die niet door gebruikers aangegeven.
Tenslotte nog iets over omgevingen waar gewerkt wordt met prototyping of vierde generatie talen.
Het werken met prototyping kan worden beschouwd als het uitvoeren van een kompleet, maar miniprojekt tijdens het ontwerp. Bij prototyping wordt immers een deel van het systeem ontworpen, gebouwd, getest en ingevoerd. De invoering heeft het karakter van een demonstratie aan gebruikers. Het iteratieve aspekt van ontwerpen kan hier volledig tot z'n recht komen. Nadat het ontwerp op deze manier heel konkreet is geworden voor gebruikers wordt het ontwerp afgerond en begint de rest van de projektaanpak. Het zal duidelijk zijn dat de bemanning van prototyping-projektgroep er anders uitziet dan van een normale projektgroep (35.2). Zie verder de aparte paragraaf over dialoogsimulatie en prototyping.
Het gebruik van vierde-generatietalen ligt in sommige opzichten dichtbij prototyping. Door de high level language leveren beide methoden snel een resultaat dat redelijk gemakkelijk is aan te passen. Bij vierde-generatietalen gaat het echter niet alleen om het ontwerpproces, maar ook om de bouw van het uiteindelijke systeem. Aanpassingen van een systeem dat is gebouwd met een vierde generatietaal gaat dan wel sneller dan van COBOL-systemen, maar in de praktijk zal er toch gewerkt worden met releases en er blijven aanpassingen voorkomen die het hele systeem betreffen en dus arbeidsintensief zijn. Het gevaar van het gebruik van vierde generatietalen is dat het ontwerp nog sterker onderschat zal worden dan nu het geval is, dat systemen nog gemakkelijker en ondoordachter zullen worden aangeschaft. Bij vierde-generatietalen ligt het aksent op het tempo waarmee projekten kunnen worden gerealiseerd. Waarschijnlijk krijgen performance-beheerders er een

groot en ondoorzichtig probleem bij! Zowel bij prototyping als bij gebruik van vierde generatietalen blijft transaktie-ontwerp nodig en mogelijk vanwege de inbreng van de gebruikers en de kwantificering ervan.

31.3 Systeemontwikkelingsmethoden en witte vlekken

In het kader van het onderwerp kunnen we van de meeste ontwikkelingsmethoden het volgende vaststellen:

- Ze dekken niet het gehele trajekt van de fase informatiebeleid tot de fase evaluatie van het ingevoerde systeem.
- Als we het geheel van de filosofie, aktiviteiten, methoden, gereedschappen en dokumentatie per fase de breedte van de systeemontwikkeling noemen dan betreffen vele methoden slechts een deel van die breedte.
- Als we letten op de resultaten van systeemontwikkelingsmethoden, dan blijkt dat er uiteindelijk altijd twee soorten produkten worden opgeleverd: programma`s en databases of bestanden. Beide ontstaan als logisch resultaat van analyse, ontwerp en bouw. Er bestaat een duidelijke relatie tussen de geanalyseerde bedrijfssituatie en de uiteindelijke bestanden op schijf. Dat blijkt niet te gelden voor de konfiguratie en het netwerk. Beide zijn uiteindelijk aanwezig, maar in geen enkele systeemontwikkelingsmethode zijn ze resultaat van bijvoorbeeld het technisch ontwerp.
- Een indeling in fasen is altijd aanwezig.
  Meestal gaat het om de volgende fasen:
  - Analyse of definitiestudie (10) of funktionele decompositie (9) of systeemonderzoek (11).
  - Logisch of funktioneel ontwerp.
  - Technisch ontwerp.
  - Bouw.
  - Tests.
  - Invoering.
  - Produktie en evaluatie.

Een belangrijke vraag blijkt te zijn: Hoe hoog zijn de muren tussen de fasen?

Er bestaan enige praktische problemen rond de scheiding tussen analyse en logisch ontwerp en die tussen logisch en technisch ontwerp. In het kader van dit boek gaat het om interaktieve toepassingen, kommunikatie met de gebruiker en netwerkontwerp. De analysefase en de fase van logisch ontwerp zijn de fasen waarin de kommunikatie met de gebruiker plaatsvindt. Het technisch ontwerp is gericht op de gekozen of de te kiezen hardware, het logisch ontwerp is er onafhankelijk van. Toch kunnen hardware- of software-beperkingen konsekwenties hebben voor de afspraken met

de gebruikers. Terugkoppeling moet dus altijd mogelijk zijn. We zullen in de volgende paragrafen het hanteren van de fasen nader bekijken. Nu reeds moet duidelijk zijn dat geen enkele weldenkende automatiseerder zal pleiten voor het afschaffen van het werken in fasen. Het blijft fout om tijdens de analyse van een probleem al in COBOL te denken. Als voor het logisch ontwerp de onafhankelijkheid van hardware karakteristiek is, mag de systeemontwerper zich ook geen zorgen hoeven maken over de implementatie. Toch is het nuttig eens na te denken over de hoogte van scheidingsmuren.

- De muur tussen analyse en ontwerp.

De scheiding tussen informatie-analyse en logisch ontwerp is duidelijk en essentieel. Tijdens de analysefase wordt de bestaande situatie in kaart gebracht. De informatie-analist kan daarom beter een materiedeskundige zijn dan een automatiseringsdeskundige. Toch wordt er tijdens het onderzoeken van de huidige situatie al gedacht aan een ontwerp. De meeste methoden die zich bezighouden met deze fase vermelden aktiviteiten als: maak een globale beschrijving van de uitvoer van het nieuwe systeem, maak een globale beschrijving van de funkties die verricht moeten worden om tot die uitvoer te komen, bepaal mogelijke hulpmiddelen en oplossingen, bepaal relevante alternatieven in termen van organisatorische vereisten, benodigde apparatuur en progammatuur (10). De bekende kosten/baten analyse komt altijd voor.

Kortom, er wordt geanalyseerd, maar af en toe wordt er wel degelijk vooruitgekeken naar een ontwerp.

Aan deze algemeen geaksepteerde werkwijze zouden we het volgende willen toevoegen. Een goede informatie-analist moet in gedachten een geautomatiseerde, interaktieve realisering van de huidige situatie voor zich kunnen zien. Als hij een gebruiker een overzicht ziet bestuderen dat uit 20 pagina's bestaat, moet hij zich een beeldscherm voor kunnen stellen met een bladerfunktie. Als hij zich verdiept in de informatie op dat overzicht, moet hij zich voor kunnen stellen hoe een computer zo'n overzicht samenstelt. Wanneer hij tijdens die meditatie zou vaststellen dat daar wel eens een lange verwerkingstijd voor nodig zou kunnen zijn, moet hij zich verder verdiepen in de huidige situatie. Is het bestaande overzicht te splitsen in delen, omdat de gebruiker per geval maar een fraktie van het overzicht gebruikt? Kan de gebruiker aangeven waar die gevallen in verschillen en waarom?

Wanneer de analysefase te grof en te vlot wordt afgewerkt, wordt de basis gelegd voor de onbegrijpelijk lange responsetijden. Gezien de tijd die er meestal beschikbaar is voor deze fase in vergelijking met de ontwerp- en bouwfase, is het geen wonder dat responsetijden van een minuut of langer in de praktijk regelmatig voorkomen. Het is daarom van belang de muur tussen analyse en ontwerp niet te hoog op te trekken. Zowel vanuit het logisch als

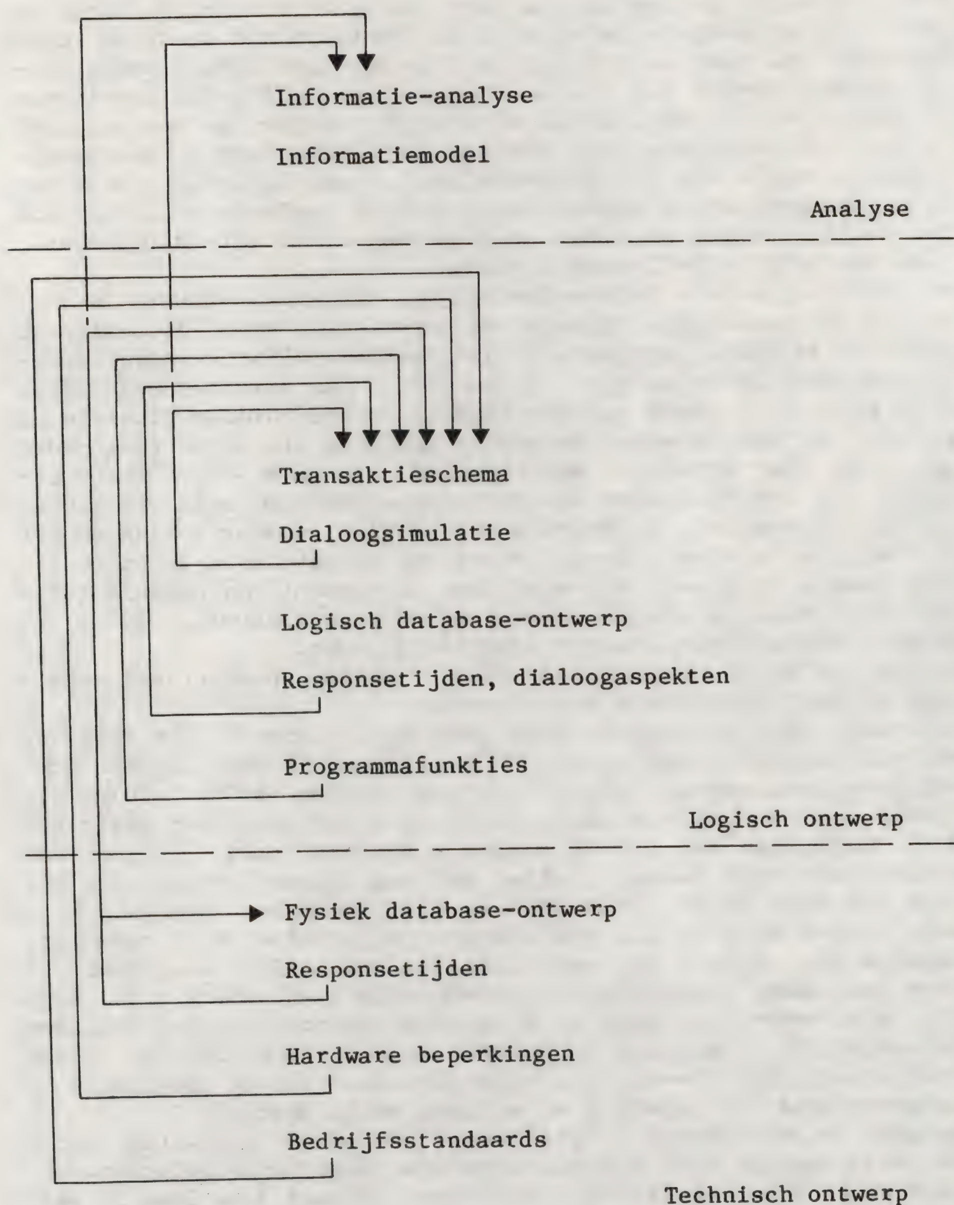


Fig 31.2 Iteraties en transaktie-ontwerp

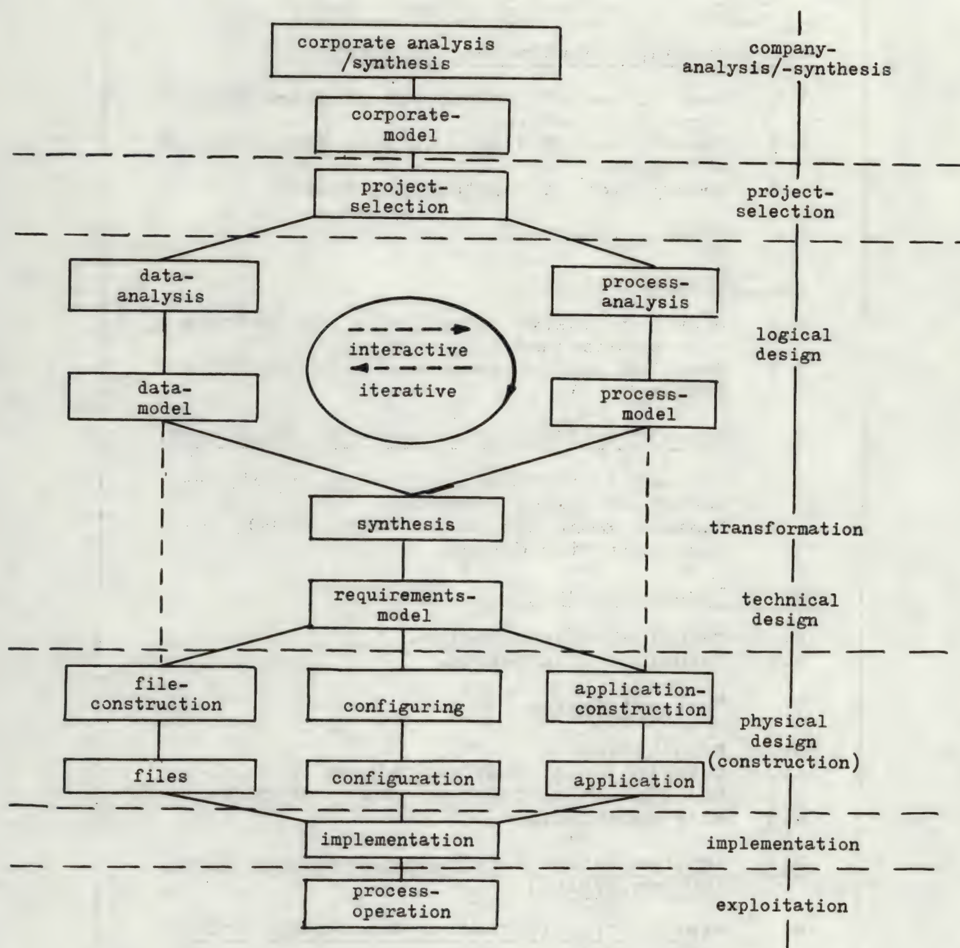


Fig 31.3 Analyse/synthese volgens de databaseclub.

Evaluation matrix, phases (rows).

1. Company analysis and -synthesis.

1.1 Overall analysis/synthesis of processes and process structures.
1.2 Overall analysis/synthesis of data and data structure.
1.3 Description of the "company model".

2. Projectselection.

3. Logical design.

3.1 Detailed analysis/synthesis of processes ---> process model.
3.2 Detailed analysis/synthesis of data ---> data model.

4. Transformation logical -> technical design.

4.1 Logical system design (process model) ---> technical system design.
4.2 Logical data structure (data model) ---> technical file design.

5. Technical design.

5.1 Technical system design.
5.2 Technical file design.

6. Physical design.

6.1 Programming.
6.2 Description of user procedures.
6.3 Description of computer centre procedures.
6.4 File construction.

7. Implementation (including acceptance test and conversion)

8. Exploitation (including maintenance and evaluation)

Fig 31.4 Fasenindeling volgens de databaseclub.

het technisch ontwerp moet een terugkoppeling mogelijk zijn naar de analysefase. In de paragraaf Responsetijden wordt dit aspekt ook genoemd.

- De muur tussen logisch en technisch ontwerp.

In sommige bedrijven is deze afscheiding onoverkomelijk hoog. Analyse en logisch ontwerp worden door een andere funktionele eenheid gedaan dan het technisch ontwerp. Bij grotere bedrijven is de reden daarvoor duidelijk. Er is een technisch koncept gekozen, nieuwe toepassingen moeten gebouwd worden op bestaande hardware en gebruikmaken van vierde generatie talen, gekozen software pakketten voor databasehandling, programmastandaards, mapping en dergelijke. Met andere woorden, het technisch ontwerp is heel duidelijk verbonden met het rekencentrum, het logisch ontwerp met de gebruikers. Soms gaat men dan ook zover dat de informatie-analisten in de gebruikersorganisatie worden ondergebracht. De kommunikatie tussen beide groepen verloopt dan ook volgens het principe van eenrichtingsverkeer. De informatie-analisten leveren hun ontwerp af aan de systeemontwerpers, die zorgen voor de realisering van de ontwerp of van iets wat er enigzins op lijkt.

In het hoofdstuk Transakties wordt aangegeven dat het transaktieschema en de schermlay-out moeten worden gekontroleerd op een aantal aspekten. Het transaktie-ontwerp moet bijvoorbeeld in overeenstemming zijn met de geldende standaards. Wanneer de standaards ontworpen, aangepast en beheerd worden door de systeemontwerpers, kan van de informatie-analisten niet verwacht worden dat hun ontwerpen altijd kloppen met de standaards. Zo kan men bijvoorbeeld gekozen hebben voor een systeemontwikkelingsmethode waarin de dialoogstruktuurdiagrammen worden gemaakt tijdens het technisch ontwerp. Een reden daarvoor is de invloed van de standaardisatie op de dialoogstruktuur: standaard funktietoetsen, standaard helpschermen, standaard afhandeling van menuschermen, enzovoort. Het innige verband tussen transaktie-ontwerp en dialoogstruktuur is duidelijk. Het is zeer onpraktisch ze door een onoverkomelijke muur te scheiden. Terugkoppeling moet mogelijk zijn zoals is aangegeven in Fig 31.2. Sommige vierde generatie talen, leggen beperkingen op aan het dialoogontwerp. Als de informatie-analist daarvan geen idee heeft, loopt hij de kans een dialoog te ontwerpen die niet gebouwd kan worden. Hij moet over de muur heen kunnen kijken, ter voorkoming van overbodige iteraties.

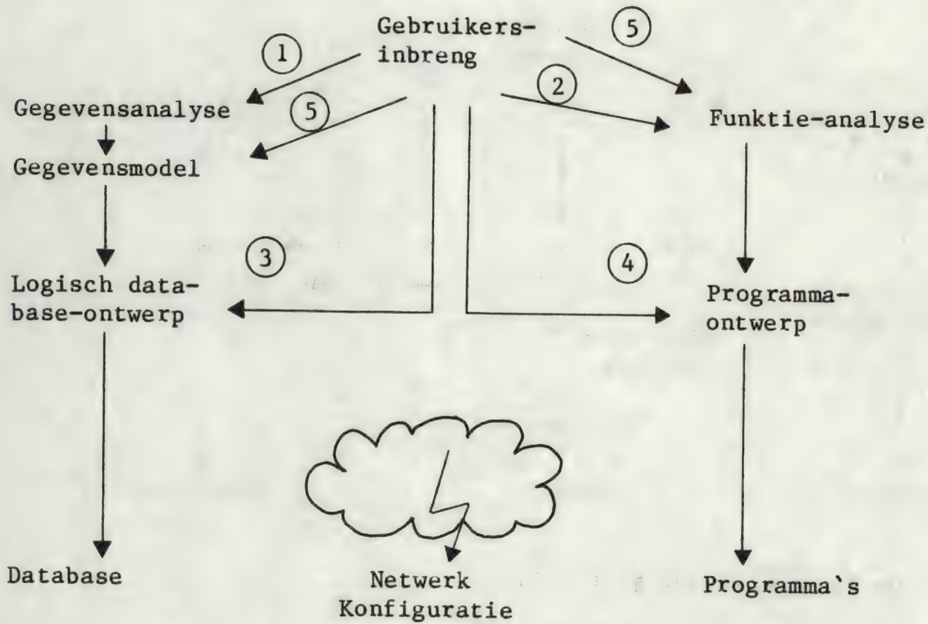
- De witte vlekken in systeemontwikkelingsmethoden.

Het doel van systeemontwikkelingsmethoden zou moeten zijn: van een vooronderzoek komen tot de bouw en invoering van informatieverwerkende systemen. De meeste methoden dekken niet het hele trajekt. De databaseclub van het NGI heeft een onderzoek gedaan naar de reikwijdte van een aantal systeemontwikkelingsmethoden

(6). Opvallend is daarbij het feit, dat in het algemene plaatje, zoals weergegeven in Fig 31.3, de konfiguratie nog voorkomt in het fysieke ontwerp (konstruktie). Daar zou het netwerk onder moeten vallen. Tijdens het fysiek ontwerp, waar programma's en bestanden worden gebouwd moet ook een netwerk worden gebouwd. Bestandsbouw is gebaseerd op het gegevensmodel, applikatiebouw op het procesmodel. De konfiguratie valt uit de lucht, wordt verondersteld aanwezig te zijn, maar is in ieder geval nergens op gebaseerd. In Fig 31.4 is aangegeven welke aspekten per fase in iedere methode zijn onderzocht op aanwezigheid: de Y-as van de evaluatiematrix. In die lijst komt de konfiguratie helemaal niet meer voor. Dat beeld is korrekt voor de onderzochte methoden. In geen van die systeemontwikkelingsmethoden bestaat een lijn van logisch ontwerp via technisch ontwerp naar topologie en dimensionering van het netwerk.
Daarnaast bestaat er nog een aantal methoden dat pretendeert een aanpak te bieden voor netwerkontwerp, maar nergens echt konkreet aangeeft hoe het moet. Hoewel termen als number of transactions, transactions usages, traffic volumes in transactions per hour, response time requirement, feedback with users, het beste doen vermoeden, leiden ze geen van allen ergens toe. In Fig 31.5 is de "normale" aanpak nog eens weergegeven. In Fig 31.6 is de werkwijze in beeld gebracht van degenen die nog iets doen aan kwantiteiten, data distribution en dergelijke. Tenslotte is fig 31.7 de komplete, in dit boek aangegeven methode. Samengevat kunnen we vaststellen dat iedereen het er over eens is dat er een relatie moet bestaan tussen wat gebruikers met beeldschermen willen en het netwerk. Daarom is de volgorde van de hoofdstukken van dit deel in grote lijnen:

- Gebruikers.
- Transaktie-ontwerp.
- Transaktie analyse.
- Netwerkontwerp.

De toegepaste methoden worden waar nodig, gekoppeld aan de methoden die gebruikt worden om uiteindelijk te komen tot programma- en gegevensontwerp. In het hoofdstuk Netwerkontwerp zullen we de werkwijze, beschreven in de vakliteratuur, bespreken.
Uit Fig 31.6 blijkt nu al dat transaktie-ontwerp en Transaktie analyse geen enkele aktiviteit van welke systeemontwikkelingsmethode dan ook vervangen. Beide methoden sluiten aan op de bekende, bestaande methoden voor programma- en database-ontwerp en leveren een bijdrage in het netwerkontwerp. Een bijdrage, omdat er aan een netwerkontwerp nog andere eisen ten grondslag liggen dan alleen het verkeer.
De tweede witte vlek in systeemontwikkelingsmethoden betreft de kommunikatie tussen gebruikers en automatiseerders. In iedere



1. Om welke gegevens gaat het?

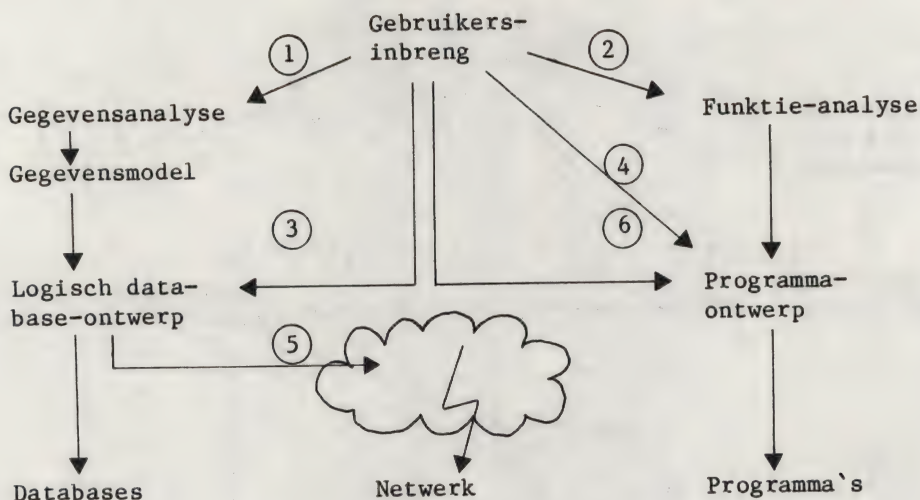
2. Wat doet de gebruiker ermee?

3. Kwantiteiten worden soms meegenomen in het database-ontwerp om toegangswegen te optimaliseren.

4. Kwantiteiten beinvloeden de keus tussen bv. batch en on line.

5. Welke waarde hebben getallen precies, behalve een relatieve waarde?
   Wat verandert er in de databasestruktuur of in het programma-ontwerp wanneer niet 1000, maar 1200 orders per dag worden verwerkt?

Fig 31.5 De meest toegepaste methode



1. Om welke gegevens gaat het?

2. Wat doet de gebruiker ermee?

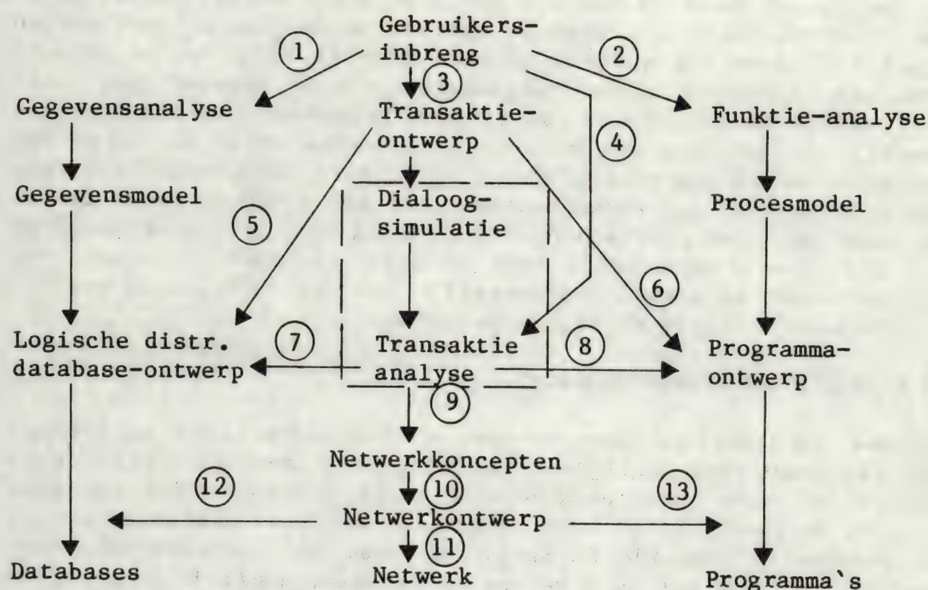
3. Kwantiteiten worden soms meegenomen in het database-ontwerp om toegangswegen te optimaliseren.

4. Kwantiteiten bepalen soms de keus tussen batch en on line.

5. Verkeer tussen databases wordt uitgedrukt in messages per hour waarbij een message niet gedefinieerd is of in traffic volumes in transactions per hour, waarbij het gaat over transactions on databases (7).

6. Hoe wil de gebruiker het op zijn scherm zien?

Fig 31.6 De methode waarin netwerkontwerp gepretendeerd wordt.



1. Om welke gegevens gaat het?
2. Wat doet de gebruiker ermee?
3. Hoe wil de gebruiker werken met een beeldscherm en hoe past dat in zijn overige taken of procedures?
4. Hoe vaak doet hij ....., hoe vaak komt het voor dat .....?
5. Vergelijking tussen gebruikte gegevens en het transaktie-ontwerp.
6. Vergelijking tussen funkties in model en in transaktie-ontwerp, realisering van de dialoog.
7. Responsetijdeisen kunnen het ontwerp beinvloeden.
8. Responsetijdeisen kunnen het ontwerp beinvloeden.
9. Op basis van de kwantiteiten, zie 4, worden koncepten ontwikkeld en doorgerekend.
10. Netwerkkoncept is gekozen, ontwerp wordt gemaakt.
11. Bouw van het netwerk.
12. Laatste afstemming over responsetijden.
13. Laatste afstemming over responsetijden.

Fig 31.7 De komplete aanpak.

projektaanpak staat natuurlijk dat gebruikers moeten worden geinterviewed in de analysefase en dat zij de in- en uitvoer moeten bepalen tijdens het ontwerp. Gezien door de bril van de ontwerpers van de systeemontwikkelingsmethoden is er helemaal geen witte vlek: er is aangegeven wat er moet gebeuren. Daarbij gaat men voorbij aan het onvermogen van een gebruiker zich op basis van papier of zelfs een schermlay-out op de buis, een beeld te vormen van zijn toekomstige werksituatie, laat staan van de kwantitatieve aspekten ervan, uitgedrukt in transakties en beeldschermuren per dag. Die kommunikatie moet volgens methoden verlopen, die kunnen worden toegepast onafhankelijk van beschikbaarheid van systemen en software en van de omgeving.

31.4 Vijf soorten vakmanschap

Wanneer interaktieve toepassingen worden doorgelicht in verband met te lange responsetijden, blijkt steeds weer dat iedere specialist op zijn terrein zijn best heeft gedaan om tot een goed ontwerp te komen. De informatie-analist heeft de gebruikerswensen zo nauwkeurig mogelijk in kaart gebracht, de database-ontwerper heeft de toegang tot de gegevens zo optimaal mogelijk gemaakt, en de programma-ontwerper heeft gelet op een goede programmastruktuur. Een systeemspecialist heeft het geheel aangepast aan de eigenschappen van het systeem, de datakommunikatiespecialist heeft in overleg met leveranciers van datatransmissie-apparatuur een netwerk opgezet. Kortom, in het geval van te lange responsetijden zijn er genoeg mensen die naar elkaar kunnen wijzen. Ieder heeft op zijn terrein z`n best gedaan, maar het resultaat is niet akseptabel voor de gebruiker. De responsetijd is alleen te beheersen wanneer in iedere fase van het projekt de betrokken specialisten worden afgerekend op te verwachten responsetijden. Over het geheel genomen zijn de volgende vakmensen daarbij nodig:
- De informatie-analisten. Zij moeten de gebruikerssituatie detailleren tot op een niveau waar ze transakties zien ontstaan. Zij moeten alternatieven voor de gebruiker kunnen bedenken, wanneer in de analysefase al duidelijk wordt dat de gestelde eisen nooit gehaald kunnen worden.
- Systeemontwerpers of technische ontwerpers. Zij vertalen het logisch ontwerp naar een technisch ontwerp en hebben voldoende kennis van bijvoorbeeld de database-software om een fysiek database-ontwerp te maken. Ze kennen de funkties, eigenaardigheden en beperkingen van een teleprocessingmonitor en weten wat wel en niet kan met de beschikbare beeldschermen.
- Systeemspecialisten. Soms zijn ze medewerkers van de computerleverancier, soms externe adviseurs, soms behoren ze tot de automatiseringsafdeling van het bedrijf. Ze hebben gedetailleerde

kennis van hardware en systeemsoftware. Ze weten bijvoorbeeld precies hoe de lijnprocedure werkt en hoe de accesmethode voor datakommunikatiepoorten optimaal gebruikt kan worden.
- Transaktie-analisten. Zij hebben met behulp van Transaktie analyse de transakties in kaart gebracht onder andere qua responsetijden en verkeer.
- Datakommunikatiespecialisten. Zij ontwerpen in samenwerking met transaktie-analisten het netwerk.
- Projektleiders of performance-beheerders. Zij koppelen in de diverse fasen deze soorten vakmanschap. Ze hoeven niet de specialistische kennis van elk genoemd vakmanschap te hebben, maar ze moeten het resultaat van ieders inbreng wel kunnen beoordelen en daarmee een idee krijgen van het totaal. Het gaat daarbij niet om algemeenheden over omstandigheden die responsetijden kunnen beinvloeden en evenmin om de berekening van responsetijden op 10 of 50% nauwkeurig.

Tijdens het logisch ontwerp wordt de verwerking van iedere interaktie door de computer tijdens de logische Transaktie analyse in kaart gebracht op basis van het gegevensmodel of het datamodel. De informatie-analist geeft per transaktie aan wat de kritische responsetijden zijn. De ontwerper van het gegevensmodel geeft aan of die tijden gehaald kunnen worden op basis van het aantal I/O`s dat nodig zou zijn volgens het gegevensmodel, er van uitgaand dat een I/O 0,1 seconde kost. Voor die gevallen waarin de responsetijden duidelijk zullen afwijken wordt overwogen of een ander toegangspad mogelijk is, ook al zou dat enige redundancy opleveren. Als dat niet mogelijk is, wordt overleg gepleegd met de ontwerpers van de fysieke database. Sommige databasemanagementsystemen bieden mogelijkheden om de fysieke opslag zodanig te besturen dat het aantal I/O`s minimaal wordt. Wanneer ook dit niet tot de gewenste resultaten leidt, worden de verwachte responsetijden besproken met de gebruiker, door ze in te stellen op de dialoogsimulator en gebruikers de transakties te laten uitvoeren. Dan kunnen de gebruikers kiezen uit drie mogelijkheden:
- aksepteren,
- een andere transaktie voorstellen,
- afzien van de transakties.

Nogmaals, het gaat er niet om responsetijden te berekenen op 10 of 50% nauwkeurig, het gaat om situaties waarin de verwachte responsetijden 200%, 300% of meer boven de gestelde eisen liggen.

Tijdens het technisch ontwerp, wanneer de fysieke database is ontworpen en in CxN-omgevingen ook het netwerk bekend of ontworpen is, wordt de technische Transaktie analyse uitgevoerd. De kritische responsetijden worden weer onder de loep genomen. De database-ontwerpers geven aan hoeveel I/O`s er moeten worden gedaan. De gemiddelde tijd voor een I/O wordt weer op 0,1 seconde

gesteld, tenzij de computerleverancier of de leverancier van het databasemanagementsysteem andere tijden aangeeft. Natuurlijk blijft iedereen volhouden dat dit getal van zoveel faktoren afhangt en dat er geen waarde voor is vast te stellen. Dat is ook zo, maar gemiddeld over een dag is de tijd per I/O groter dan nul en kleiner dan 1 seconde. De uiterste grenzen zijn dus bekend. Met Transaktie analyse is het simpel om even twee verschillende waarden toe te kennen aan de tijd per I/O en de resultaten te beoordelen. We gaan hierbij natuurlijk niet uit van een overbelaste computer. Het doel van dit soort methoden is onder andere de computerleveranciers te konfronteren met de cijfers van Transaktie analyse en op die manier tot verantwoorde konfiguraties te komen. De datakommunikatiespecialist geeft aan in hoeverre het op basis van Transaktie analyse ontworpen of beschikbare netwerk de responsetijden beinvloedt. De transaktie-analist kontroleert dat aan de hand van de resultaten van Transaktie analyse.
De systeemspecialist levert gegevens over de eigenschappen van het operatingsystem ten aanzien van I/O voor het laden van programma`s, de prioriteiten, de lijnbesturing enzovoort.
Wanneer nu blijkt dat er toch niet voldaan kan worden aan de eisen van gebruikers, dan is dit het laatste moment om ze in te schakelen en problemen achteraf te voorkomen. Ze mogen dan weer kiezen uit de drie genoemde mogelijkheden. Een operationsmanager die een onderzoek instelt naar de oorzaak van de lange responsetijden van bepaalde toepassingen krijgt vaak te maken met die rapporten van de verschillende specialisten. Dan zal hij achteraf hetzelfde moeten doen als wat er tijdens het ontwerp had moeten gebeuren, maar wat niet gebeurd is, omdat het veel gemakkelijker is iedereen zijn deeltje van het systeem te laten bouwen zonder zich te bekommeren om de performance van het geheel. En als hij met die verschillende rapporten niet verder komt, schrijft hij een algemeen rapport over het probleem van het beheer van performance en de aanschaf van een aantal uitbreidingen, meestal op advies van de leverancier. Het is alleen jammer, dat al zoveel bedrijven hun systeem hebben uitgebreid, zonder garanties ten aanzien van de te bereiken verbeteringen, en met minimaal resultaat. Het uitbreiden van het geheugen of de schijfkapaciteit heeft natuurlijk ook weinig zin als het knelpunt in het netwerk zit. Er is een performance-beheerder nodig die alle aspekten van de responsetijden in de gaten houdt. Hij kent alle mogelijkheden van beschikbare meet-tools en is betrokken bij het ontwerp van nieuwe toepassingen zoals bovenstaand is beschreven.

31.5 Aanpak

Als het taktisch automatiseringsmanagement besluit een methode

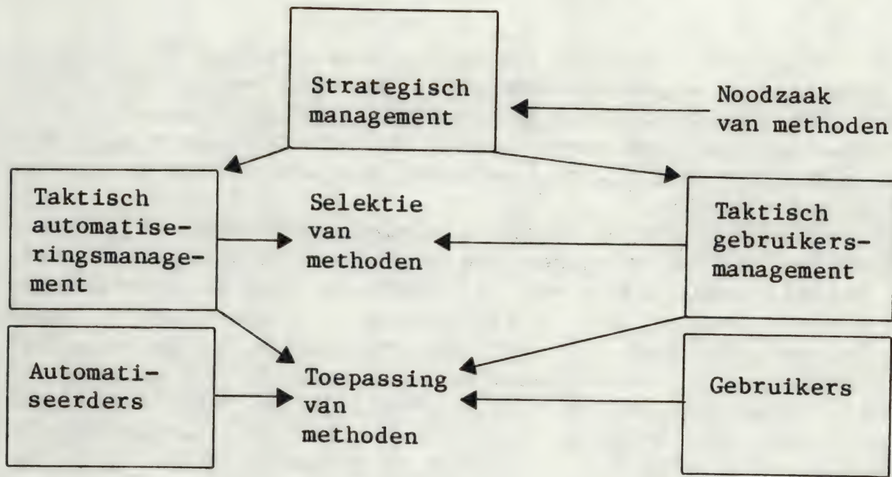


Fig. 31.8 Aanpak

als transaktie-ontwerp in te voeren, dan zou wat hen betreft alleen nog maar gewacht hoeven te worden tot de gebruikers de methoden gaan eisen. Deel 1 en deel 2 zouden dat effekt moeten hebben. Transaktie-ontwerp is immers in het belang van gebruikers. Helaas zal het vaak zo niet werken. Automatiseerders zullen het eerst het effekt van dit soort methoden goed kunnen schatten. Zij zullen ook begrijpen dat tevreden gebruikers uiteindelijk in hun eigen belang zijn. Daarom zullen zij genegen moeten zijn gebruikers te verkopen wat ze eisen. Naast tevreden gebruikers aan het eind van ieder projekt, hebben automatiseerders ook veel belang bij gebruikers die tijd hebben gepland en gegevens willen verstrekken tijdens ieder projekt. Dat is dan de tweede reden om iets te doen aan invoering van methoden in de gebruikerswereld. Zelfs al zouden strategen en taktici onder de gebruikers na de lezing van deel 1 respektievelijk deel 2 overtuigd zijn van het nut van de methoden, dan nog zullen automatiseerders als trekkers moeten fungeren.
Een en ander is in beeld gebracht in Fig 31.8. Een presentatie van een interne of externe adviseur over de noodzaak van methoden moet leiden tot de strategische beslissing dat gebruikers worden ingeschakeld als mede-ontwerpers, dat daarvoor een planning gemaakt moet worden en dat projekten in delen in opdracht worden gegeven. Het eerste deel loopt tot en met het logisch ontwerp. Tijdens de presentatie kan men de toehoorders verwijzen naar de

diverse delen van dit boek. Vervolgens moeten taktische automatiseringsmanagers en gebruikersmanagers het eens worden over de methoden. Een proefprojekt is daarvoor een uitstekend middel. Om te voorkomen dat dat niet werkt, omdat de methoden nieuw zijn voor zowel gebruikers als automatiseerders, kan een externe deskundige worden ingeschakeld.
Na de evaluatie dient te worden vastgelegd hoe de dokumenten worden ingepast in de bestaande projektaanpak. De automatiseringskant daarvan wordt uitgebreid behandeld in deel 4. Voor gebruikers bestaat meestal geen projektaanpak. Zij ontvangen de gegevens ten aanzien van de ontworpen transakties in de vorm van transaktieschema's en schermlay-outs. Daarmee is het gebruik van methoden te managen, dat blijft noodzakelijk. Het zou niet de eerste keer zijn dat volgens informatie-analisten dialoogsimulatie is uitgevoerd terwijl bij oplevering de gebruiker achteroverslaat als hij zich realiseert hoe de procedure aan het beeldscherm in elkaar zit: de informatie-analist heeft de simulator alleen gebruikt om schermlay-outs te laten zien!

Hoofdstuk 32
Gebruikers

32.1 Wie zijn de gebruikers?

We hebben ze steeds aangeduid met: alle niet-automatiseerders. In deel 1 hebben we gebruikers ingedeeld op basis van hun inbreng in het ontwerpproces. Voor het taktisch automatiseringsmanagement is een andere indeling nuttig. Als we even de rangen en standen buiten beschouwing laten, kunnen we drie soorten gebruikers onderscheiden:

- gebruikers die alleen geinteresseerd zijn in de kosten en de opleveringsdatum
- gebruikers die de funkties van het informatieverwerkende systeem moeten vaststellen
- gebruikers die zullen werken met de beeldschermen.

Soms vallen de drie types samen in een groep gebruikers. In dat geval ondergaan ze drie keer een verschillende behandeling.
Gebruikers van het eerste type zijn meestal strategische of taktische managers. Naast alle andere dingen die besproken zijn moet ze duidelijk gemaakt worden wat het belang is van een projektaanpak en wat de verschillende fasen voor de gebruikers betekenen. Per projekt moet voor iedere groep gebruikers een presentatie worden gehouden over projektaanpak, de samenwerking tussen gebruikers en automatiseerders per fase, toepassingen van methoden en gereedschappen, de benodigde tijd van de gebruikers en de be-

nodigde voorbereiding. Het moet het eerste type gebruikers duidelijk worden dat het schatten van een opleveringsdatum na een vooronderzoek geen waarde heeft. Automatiseerders die klagen over de kommunikatie met de gebruikers van welk type dan ook, maar nog nooit een presentatie hebben gehouden met overhead-foils of flip-over, moeten niet zeuren. Problemen die automatiseerders met gebruikers hebben worden vaak veroorzaakt door de automatiseerders. Gebruikers moeten een projekt in fasen in opdracht geven. Door toepassingen van methoden als dialoogsimulatie en Transaktie analyse heeft het ook zin om dat te doen. Tijdens het logisch ontwerp krijgt de gebruiker immers een konkreet inzicht in de toekomstige situatie.
Gebruikers van het tweede type zijn belangrijke gesprekspartners voor informatie-analisten en systeemontwerpers. Zij zullen immers een duidelijke inbreng hebben in de diskussie over funkties die het systeem moet verrichten, gegevens die daarbij nodig zijn, koppelingen van systemen ten dienste van de centrale verwerking of de kommunikatie tussen systemen. Zij hebben meestal een duidelijke inbreng in gesprekken over kosten en baten van verschillende technische oplossingen. De kwantitatieve aspekten van het ontwerp zoals Transaktie analyse die oplevert zal hen in sterke mate interesseren.
Deze gebruikers zullen ook vaak optreden als eigenaars van gegevens. Zij stellen vast om welke entiteit het gaat, wat de attributen zijn en welke gebruikers het gegeven mogen aanmaken, opvragen, wijzigen of verwijderen. De diskussie rond het eigenaarschap van gegevens valt buiten het kader van dit boek, maar bij het ontwerpen van transakties door gebruikers wordt er stilzwijgend van uitgegaan dat de gebruikers en informatie-analisten zich aan de regels houden of dat er een groep is die alle transaktieschema`s kontroleert. Het feit dat transaktieschema`s gebruikersdokumenten zijn is daarbij een voordeel want nu kan de kontrole worden uitgevoerd door gebruikers (eigenaars).
De gebruikers van het laatste type zijn belangrijk voor het sukses van een projekt. De meeste systemen funktioneren wel overeenkomstig de gestelde eisen. De problemen liggen op het niveau van het transaktie-ontwerp en dan gaat het over
- de akseptatie van de nieuwe manier van werken
- de problemen rond het bedienen van beeldschermen, de dialoog
- de performance problemen
- de aansluiting op de handmatige verrichtingen
- enzovoort.

Deze problemen zijn volledig op te lossen door dit gebruikerstype te laten optreden als mede-ontwerpers van alle transakties.

32.2 Problemen laten waar ze horen

Achterafpraters hebben meestal gelijk. Wanneer een projekt is gerealiseerd en de gebruikers enige tijd gewerkt hebben aan de beeldschermen, is het leveren van kritiek een eenvoudige zaak. Moeilijker is het om tijdens analyse- en ontwerpfase de verantwoordelijkheden goed vast te leggen.
De automatiseringsafdeling is niet verantwoordelijk voor alles wat er mis kan gaan. Sommige problemen horen thuis op de gebruikersafdelingen, andere op de automatiseringsafdeling. Laten we als voorbeeld eens nemen de dimensionering van een systeem. Uiteindelijk zal de grootte van een systeem afhangen van het gebruik. Zo zal het aantal beeldschermen op een afdeling afhangen van het aantal transakties dat de gebruiker per dag wil realiseren. In dit geval is dus de schatting van het aantal transakties van groot belang. De automatiseringsafdeling kan twee soorten fouten maken. Men kan te luchtig over het probleem heen lopen door zich bijvoorbeeld niet te verdiepen in pieksituaties en toch een systeem op poten te zetten. Wanneer dan achteraf het aantal beeldschermen niet voldoende blijkt te zijn, is de kritiek van de gebruikers terecht.
De andere soort fout die men kan maken is dat men zich verdiept in alle kwantitatieve aspekten, maar deze gegevens in de ontwerpfase onvoldoende gebruikt. Meestal is een systeem op een aantal manieren te ontwerpen. Er zijn voor veel transakties alternatieven te bedenken die konsekwenties hebben voor de manier van werken. Deze alternatieven moeten nu, via methoden uit dit boek, omgerekend worden naar konsekwenties voor de gebruiker. Wanneer dat niet gebeurt, is de kritiek van de gebruiker ook terecht. Gebeurt dat wel, dan wordt de gebruiker gekonfronteerd met de gevolgen van zijn uitspraken, tijdens het logisch ontwerp. Zonder die terugkoppeling is het toch ook begrijpelijk dat gebruikers terughoudend zijn bij het noemen van cijfers? Zij kunnen de gevolgen niet overzien.
Van heel andere aard zijn de bedrijfspolitieke problemen die gebruikers kunnen hebben. Als een afdelingschef voelt aankomen dat een kwantitatieve analyse van de hoeveelheid werk op zijn afdeling, zal leiden tot de konklusie dat hij meer mensen in dienst heeft dan noodzakelijk is, zal hij zich op allerlei manieren proberen te onttrekken aan het noemen van cijfers. In dat geval doen de automatiseerders schattingen en rekenen die om naar gevolgen voor de gebruikers. De afdelingschef wordt dus gekonfronteerd met konklusies van schattingen van de automatiseerders. Zolang de schattingen niet worden gekorrigeerd, zijn ze juist. Dat betekent dat de toekomstige situatie kwantitatief in kaart wordt gebracht, in termen van de gebruikers, zoals het aantal beeld-

schermen en het aantal beeldschermuren per dag. Met andere woorden, het projekt loopt toch wel door, maar de informatie-analisten behoren in dit soort situaties toch duidelijk te reageren in de richting van hun management. Sociale of bedrijfspolitieke problemen moeten niet worden opgelost door automatiseerders, maar wel gesignaleerd.

Hoofdstuk 33
Computerleveranciers

33.1 Commercie en techniek

Verkopen is een vak. De tijd van breedsprakige marskramers is voorbij. In de computerwereld is ook aan de verkoopkant het aureool verdwenen en de klanten zijn mondiger geworden.
Bij de verkoop van technische apparatuur doet zich altijd het probleem voor dat een goede verkoper geen technicus is en een technicus geen goede verkoper. Het resultaat van enige samenwerking tussen verkopers en technici is meestal dat zowel brochures als offertes bol staan van technische termen, terwijl de toekomstige koper op zoek is naar een oplossing voor zijn problemen. Technische specifikaties vertalen naar verkoopargumenten lukt kennelijk niet altijd even goed. Trouwens, aan inkoopzijde begeeft men zich ook nogal vaak op het gebied van vergelijkingen van irrelevante technische details zoals MIPS, channelspeed, clockfrequency en dergelijke.
Ook computersystemen worden gekocht als oplossing voor een bepaald probleem. Daarbij bevinden computerverkopers zich meestal in de komfortabele situatie dat het probleem niet of nauwelijks is vastgesteld in termen die vertaalbaar zijn naar een konfiguratie en dus naar een kostenplaatje.
Uit allerlei metingen achteraf, vaak door de leveranciers zelf uitgevoerd, blijkt namelijk dat de performance van een konfigura-

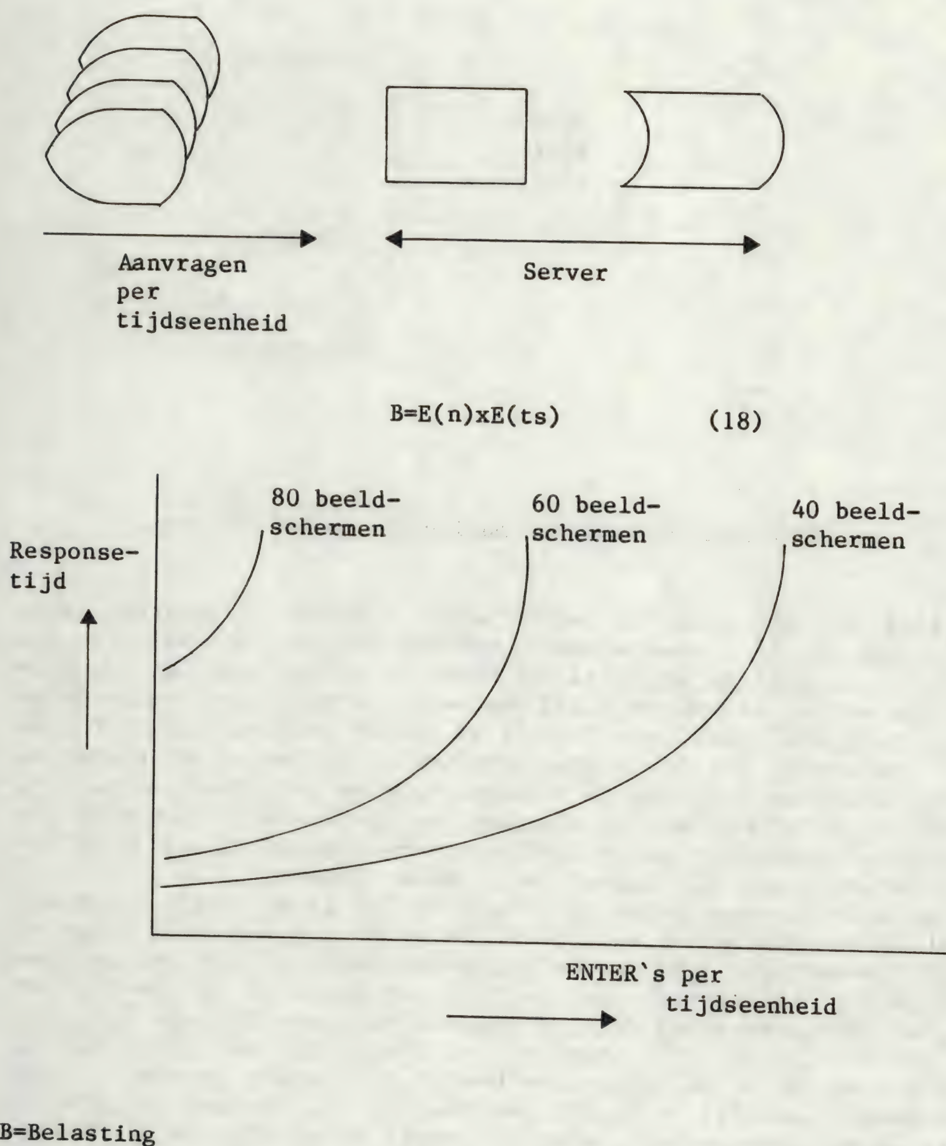
tie afhangt van zaken als het aantal ENTER`s per tijdseenheid. In het algemeen worden aanvragen voor offertes verstuurd, wanneer nog niemand daar een idee over kan hebben. Het maken van een offerte is grotendeels een zaak van kommercie. Als drie konkurrenten met een konfiguratie aankomen, die de helft kost van degene, die een verkoper in gedachten had, dan past hij zich bijzonder gemakkelijk aan. Trouwens, zijn systemen zijn toch gemakkelijk uit te breiden? Enige spreiding in de omzet hoeft niet slecht te zijn voor een verkoper. Soms stelt de inkoper eisen aan de performance in termen van gewenste responsetijden. Meestal in termen van een gemiddelde responsetijd. In de praktijk ligt zelfs geen enkele systeemontwerper er wakker van, dus waarom zou een computerleverancier zich er zorgen over maken? Hij levert een systeem dat flitsend kan werken, als er "goede" applikaties op worden gebouwd. Bij de aanschaf van systemen is daar echter nog door niemand over nagedacht met de gedetailleerdheid die computerleveranciers zeggen nodig te hebben voor de bepaling van een konfiguratie. Kortom, voor computerverkopers bestaat er niet echt een technisch probleem. Hun problemen liggen alleen op het gebied van de kommercie. Het is geen toeval dat systemengineers regelmatig vaststellen dat er te kleine konfiguraties zijn verkocht. Gelukkig voor de verkoper, bestaat er in de automatisering geen weg terug. Als er een paar toepassingen draaien, zijn de automatiseerders opgeleid en ingewerkt en is overstappen op een ander fabrikaat er niet meer bij.
Vervelend wordt het wanneer gebruikers tijdens verkoopgesprekken konkreet worden en uitspraken willen hebben over bepaalde responsetijden. Een systeemengineer kan dan al gauw de helpende hand bieden door uit te leggen waar een responsetijd allemaal van af kan hangen: de grootte van het programma, het aantal I/O`s, de schermlay-out, de definities van de velden, de transakties op andere beeldschermen, enzovoort, enzovoort. Soms wordt daarbij de methode toegepast waarbij de gebruik een aantal grafieken wordt getoond die de responsetijd weergeven als funktie van het aantal ENTER`s per uur, het aantal beeldschermen, het aantal schijven, de geheugengrootte en een mengsel van lichte, middelzware en zware toepassingen. Als de klant nu even zijn situatie aangeeft dan kan de responsetijd worden afgelezen. Dat kan hij natuurlijk niet en dus geeft de leverancier een advies op basis van zijn kennis, inzicht en ervaring. Bij het netwerkontwerp zullen we zien dat hetzelfde geldt ten aanzien van leveranciers van netwerkapparatuur.
In (34) zet een computerleverancier de aanpak op papier. Ook daar blijkt weer dat leveranciers van gebruikers gegevens verwachten die nodig zijn bij de bepaling van de konfiguratie bij aankoop, en die een gebruiker nimmer ter beschikking heeft als hij niet

eerst het transaktie-ontwerp achter de rug heeft. Enkele vragen: "Average size of input (nearest 20 characters)? Average size of output (nearest 20 characters)? Average numbers of inputs in a 10-minute period?" Andere aspekten die moeten worden beschreven: "Quantaty and type of data which must flow into and out of system
a. at each site bij hour
b. for each user group by hour
c. for each user groep bij site by hour
d. peak rates and duration of peaks by site"
In de rest van het artikel wordt gesuggereerd dat computerleveranciers met dit soort cijfers een konfiguratie kunnen bepalen. Hoe gebruikers bijvoorbeeld gemiddelde berichtlengtes zouden moeten schatten, wordt ook hier niet aangegeven. Heel vervelend wordt het als blijkt dat de inkoper en de automatiseerders in de verkoopfase al die gegevens kunnen verstrekken, omdat er tijdens het logisch ontwerp dialoogsimulatie en Transaktie analyse zijn uitgevoerd, voordat offertes werden aangevraagd. Dan moet blijken of de computerleverancier de grafieken, die hij zo hij ze al heeft, alleen bij hoge uitzondering gebruikt, ook tijdens zo`n konfrontatie in de strijd wil werpen. Leveranciers die metingen hebben gedaan aan hun systeem zouden die cijfers moeten willen leggen naast de cijfers van Transaktie analyse. Leveranciers die geen metingen hebben gedaan vallen vanzelf af.
Soms bieden computerleveranciers in de verkoopfase het gebruik van een software tool aan om de performance van een systeem te begroten. Uiteraard moet er dan een groot aantal parameters worden ingevoerd, voordat er gerekend kan worden. De waarde van dat soort simulaties wordt bepaald door de nauwkeurigheid waarmee men de interaktieve toepassingen kan beschrijven. Ook daar blijkt het onder andere te gaan om precies hetzelfde soort gegevens als die Transaktie analyse oplevert: ENTER`s per tijdseenheid, I/O`s per ENTER, etc. Meestal moet het computergebruik voor de simulatie en de begeleiding betaald worden, des te meer reden om het goed te doen.

33.2 Performance van interaktieve toepassingen

Performance-onderzoek in computersystemen blijkt een komplexe zaak te zijn. Er is een groot aantal parameters dat de responsetijden kan beinvloeden. In (29) wordt een groot aantal aspekten genoemd die van invloed zijn op de performance, waarvan de meeste technische details in de hardware betreffen waarin niemand zich in de praktijk zal verdiepen. Een lichtpunt is dat kennelijk al die aspekten maar een beperkte invloed hebben op de performance. Er bestaan allerlei methoden om een systeem te "tunen", maar de invloed daarvan is beperkt. Allerlei parameters om de performance

te besturen komen neer op het verdelen van de beschikbare kapaciteit over de toepassingen. Het verhogen van de prioriteit van de ene toepassing betekent degradatie van een andere. Als de buffers voor het databasemanagement-systeem worden uitgebreid, is er minder ruimte beschikbaar voor programma's en wordt er weer meer geswapped. Kortom, het is op detailniveau zeer komplex, maar al die parameters hebben slechts een beperkte invloed.
Een ander punt is de responsetijdmetingen. De grafieken, als in Fig. 33.1, hebben allemaal de typische vorm van de grafieken uit de wachtrijtheorie. In feite kan ieder computersysteem gezien worden als een server die een bepaalde tijd nodig heeft om een aanvraag af te handelen en dus een beperkt aantal aanvragen per tijdseenheid kan verwerken.
Er vanuit gaande dat de I/O's de beperking vormen gaat het dus om de belasting die een beeldscherm betekent voor een schijf. Dat hangt af van de dialoog (ENTER's per tijdseenheid), het aantal I/O's per ENTER en de tijd per I/O. Bij de tijd per I/O gaat het om de gemiddelde servicetijd over een uur of een dag gemeten. De gemiddelde tijd is niet nul en is kleiner dan 0,2 tot 0,5 seconden. Daarin is dus begrepen alle overhead van het databasemanagement-systeem of het filemanagement-systeem.
In (30) wordt een lans gebroken voor het uitvoeren van vergelijkende metingen van performance van systemen voor interaktieve toepassingen aan de hand van een standaardtransaktie. Transaktie analyse levert standaardcijfers op: het aantal ENTER's per tijdseenheid, het aantal I/O's per ENTER en het benodigde aantal beeldschermen.
Met de algemene bruikbaarheid van de resultaten van Transaktie analyse is nog onvoldoende ervaring opgedaan. Dat komt voor een deel door de geheimzinnigheid waarmee computerleveranciers hun cijfermateriaal omgeven. Daarnaast zijn computersystemen en database-managementsystemen in dit opzicht niet zonder meer vergelijkbaar. Als het ene systeem door de applikaties veel gemakkelijker te bedienen is of wel meer funkties biedt dan het andere, zijn de hoeveelheden I/O's en dus de belasting niet zomaar vergelijkbaar. Ten aanzien van het beheer van de performance van een operationeel computersysteem is in ieder geval duidelijk dat de systeembelastingsattributen inhoud geven aan begrippen zoals zware en lichte transakties.
Verder dient in het kader van het beheer van de performance, na iedere toevoeging van nieuwe toepassingen met een meetprogramma de gemiddelde tijd per response-eenheid (parameter 22) te worden gemeten. Dat is een meetprogramma dat gedurende enkele dagen bijvoorbeeld ieder kwartier, tien keer een meting doet van de tijdsduur van de gekozen response-eenheid op de database van de betrokken toepassing, terwijl de beeldschermen in gebruik zijn. De



B=Belasting

Fig. 33.1 Performance

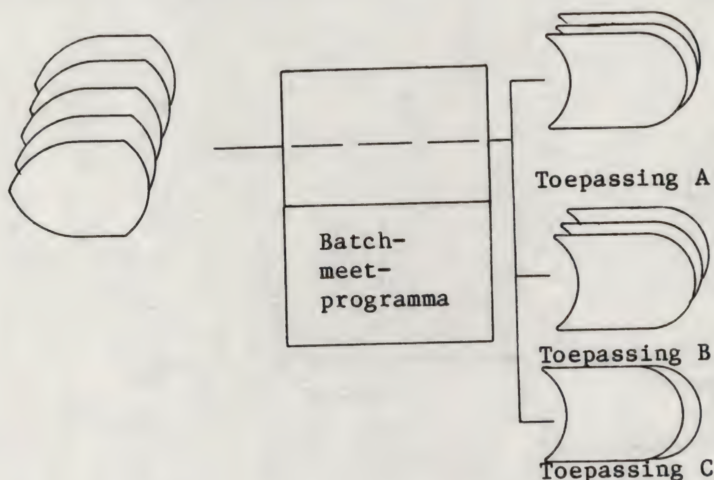


Fig. 33.2 Performance-meting per groep schijven.

trend van de gemiddelde waarde wordt bijgehouden om ervaring op te doen met de systeembelastingsattributen van transakties op het betrokken systeem en om tijdig vast te stellen dat het beruchte 70%-punt van de wachtrijgrafieken wordt bereikt. Het meetprogramma voert een specifieke I/O uit of enkele verschillende I/O`s op de database. Van de tijdmetingen wordt de gemiddelde waarde en de spreiding berekend en bewaard door de performancebeheerder. Blijkt de status van de database van grote invloed te zijn dan moeten de metingen gedaan worden op momenten dat de status vergelijkbaar is met die tijdens voorgaande metingen.
Het meetprogramma werkt per schijf of per groep schijven met applikatiebestanden of databases. Er wordt dus niet meer gelet op de transakties, maar op de gegevensverwerking door de applikaties. Per schijf of, bij multivolume files of databases, per groep schijven wordt de tijd gemeten voor een logische I/O, een databasecall of file acces of het gemiddelde van enkele I/O`s, die veel in de betrokken applikatie programma`s voorkomen. Systeemschijven blijven buiten beschouwing. Hun invloed is indirekt in de meting opgenomen, omdat bij applikatie-I/O`s ook het operating system is betrokken.
Bij voortschrijdende uitbreiding van het aantal toepassingen zal op een gegeven moment blijken dat de gemeten tijd voor alle toepassingen toeneemt. In dat geval begint het operating system of de hardware het knelpunt te worden. Op deze manier wordt perfor-

mance beheerd en kan uitbreiding van het systeem beter bestuurd worden, voordat de gebruikers beginnen te klagen. De performance van een systeem is van invloed op de responsetijden. Een overbelast systeem zal over de hele linie lange responsetijden opleveren.
Een heel ander aspekt van de verwerking, het ontwerp van de toepassing en/of de database, heeft op bepaalde responsetijden een veel grotere invloed. Dit aspekt wordt op het detailschema in kaart gebracht en geeft via een geschatte tijd per response-eenheid van 0,1 seconde tijdens het logisch ontwerp al aan of er met het ontwerp misschien iets mis is. Daarnaast kan vergroting van de waarde van de tijd per response-eenheid op het detailschema de invloed van een overbelast systeem aangeven.
Als we nu overbelaste systemen even buiten beschouwing laten, moeten we onderscheid blijven maken tussen zaken die met tuning en die met ontwerp hebben te maken. Onder tuning vallen het fysieke database ontwerp, en alle DBMS-parameters. Onder ontwerp vallen zaken als responsetijdeisen, transaktie-ontwerp, toegangspad-analyse en logisch database-ontwerp. In het deel voor de informatie-analisten worden de ontwerpaspekten verder uitgewerkt.

33.3 Het volgende performance-debâcle

Veel managers in de automatisering kijken reikhalzend uit naar de tijd dat er technieken ontwikkeld zijn die alle performance problemen van tafel zullen vegen. Geen lastige problemen meer rond tuning, opsporing van knelpunten, responsetijden en moeilijke methoden als Transaktie analyse. Glasvezels, local area networks en steeds snellere hardware moeten daar toch een keer toe leiden. Misschien is dat ook wel zo. In de dagelijkse praktijk heeft echter goedkoper wordende hardware eerder geleid tot overbelaste minicomputers dan tot minder performance-problemen. De volgende fase is die van de microcomputer. De hardware is nu zo goedkoop dat we van iedere gebruiker een computereigenaar kunnen maken. Werd de minicomputer nog ingezet als systeem voor kleine bedrijven of bedrijfsonderdelen, de microcomputer kan worden ingezet per gebruiker. Toch blijkt heel gauw dat een micro met twee floppies maar zeer beperkt bruikbaar is. Een harddisk is al een verbetering. Dan blijkt de hardware toch ook weer niet zo goedkoop te zijn dat iedere gebruiker voorzien wordt van een P.C. met een harddisk en komen we tot een netwerk van micro's met een fileserver of proberen we van de personal computer toch weer een groepscomputer te maken door aan een micro verscheidene beeldschermen aan te sluiten. Dat er echter maar een beeldhuis rechtstreeks verbonden is met het geheugen en de andere via een seriele interface, ontgaat de meeste liefhebbers. Dat bij gedeeld ge-

bruik van de hard disk bij beide oplossingen weer een performance-probleem ontstaat valt niet te ontkennen. Onlangs werd een multi-user micro aangekondigd voor dertig gebruikers. Hoe lang is het geleden dat de mogelijkheden van een minicomputer beoordeeld werden door aan de achterkant het aantal stekkers voor beeldschermen te tellen? De praktijk heeft geleerd dat er niet zoveel verband bestaat tussen het maximum aantal beeldschermen en het aantal waarbij het systeem nog redelijk werkt. Ook hier schijnt de geschiedenis zich te moeten herhalen. Een microcomputer kan via een speciale kaart of via de standaard seriele interface worden verbonden met een echte computer. Hoezeer het ook voorspelbaar is, het verzenden van bestanden duurt soms de eeuwigheid van een kwartier tot een half uur. Micro`s aangesloten aan een clustercontroller kunnen om die reden het werken aan de andere beeldschermen lange tijd onmogelijk maken. Gezien de geringe gerichte inspanning die vele bedrijven zich getroosten om de performance te beheren, valt te verwachten dat in de toekomst het aantal lokaties waar zich performance-problemenvoordoen, rechtevenredig zal stijgen met het aantal computereigenaars.
Ook deze gebruikers zullen voor hun problemen automatiseerders nodig hebben. Een informatiecentrum kan daarin een bijdrage leveren, maar niet de strukturele problemen oplossen. Na korte tijd van stand-alone-gebruik komt de behoefte aan grotere bestanden of aan gegevens die zijn opgeslagen in mainframes. Het beslissen over software op het mainframe ten dienste van microgebruikers valt meestal buiten het terrein van een infromatiecentrum. Een dergelijk centrum zou veel meer een bijdrage moeten leveren in het leren analyseren van informatiebehoeften en die vergelijken met de mogelijkheden, dan in het kiezen van een microcomputer.
Bedrijven waarvan het strategisch management besloten heeft microcomputers in te voeren omdat de automatiseringsafdelingen toch niet aan de wensen van gebruikers kunnen voldoen, zullen voor de microverkopers gewillige slachtoffers zijn.

Hoofdstuk 34
Transaktie-ontwerp

34.1 Transaktie als entiteitstype

Er is in de loop der jaren een groot aantal systeemontwikkelingsmethoden ontstaan. Vergelijkende onderzoekingen hebben aangetoond dat weinig methoden kompleet zijn. De meeste methoden leggen het aksent op een bepaald aspekt van de automatisering of ze betreffen maar een deel van de fasen die doorlopen worden. Maar ook in de meest komplete methode ontbreekt het transaktie-ontwerp in de betekenis die hier bedoeld wordt. Een transaktie is het geheel van elkaar afwisselende menselijke handelingen en computerverwerkingen, dat herhaald wordt uitgevoerd of dat door de gebruiker als een afgeronde aktiviteit wordt gezien. Een transaktie wordt vastgelegd op een transaktieschema. Daarmee zijn dus de menselijke handelingen in de systeemdokumentatie vastgelegd. Tijdens Transaktie analyse worden de transakties gekwantificeerd, het aantal per dag bijvoorbeeld, maar ook de menselijke handelingen en de verwerking door de computer.
Een transaktie is dus nu een "entiteit" geworden, waaraan eigenschappen kunnen worden toegekend, zoals aantallen, lokaties, gewenste responsetijden, doorlooptijd en systeembelasting. Daarmee is aangegeven, dat transaktie-ontwerp iets anders is dan gebruikers interviewen, beeldschermlay-outs maken, procedures ontwerpen, of problemen op het menselijke vlak identificeren en oplos-

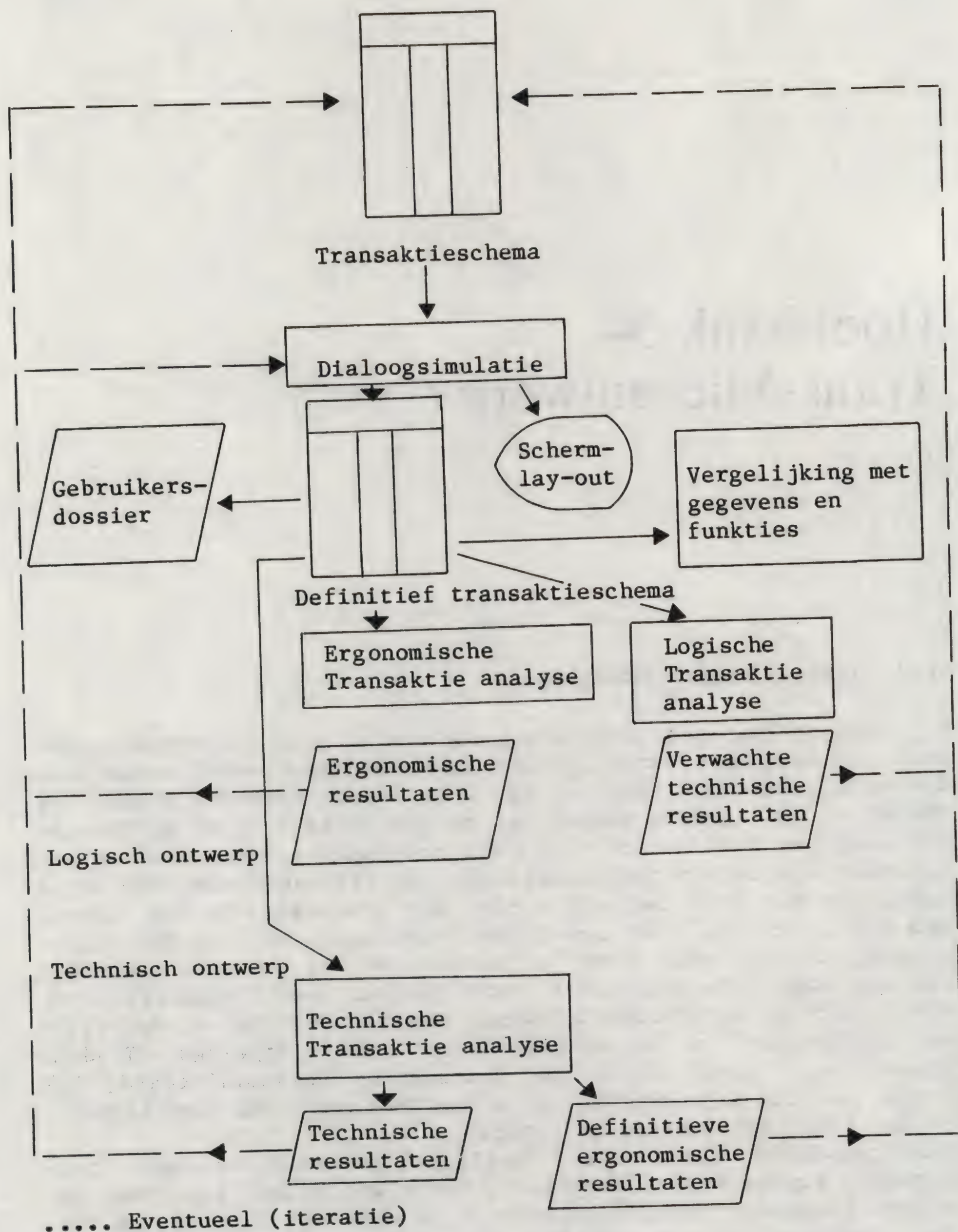


Fig. 34.1 Transaktie-ontwerp in beeld

sen. In de meeste handboeken voor systeemontwikkelingsmethoden staat hoogstens dat soort termen. "Gebruikers interviewen" is geen methode. Daarom ontbreekt elk uitzicht op goede en bovenal konstante kwaliteit van het resultaat. Van een methode zou pas gesproken kunnen worden wanneer per interview minstens is vast gelegd welke formulieren moeten worden ingevuld, en welke dokumenten worden opgeleverd. Een dokument is in dit verband iets anders dan "free format" proza. Hetzelfde geldt voor de andere termen.
Een ander aspekt van transaktie-ontwerp is de relatie met het netwerkontwerp via Transaktie analyse. Alle getallen rond transakties leiden tot getallen in het netwerkontwerp: aantallen beeldschermen, netwerkbelasting, bezettingsoverzichten per werkplek, basisgetallen voor systeembelasting enzovoort. De witte vlek in alle systeemontwikkelingsmethoden is het ontwerpen van de konfiguratie en het netwerkontwerp. In een enkele methode worden konfiguraties getekend, maar er is geen relatie met cijfers uit het logisch ontwerp. Zelfs bij James Martin is er een missing link tussen Information Engineering (7) en Systems analysis for Datatransmission (8) zoals we in een volgend hoofdstuk zullen vaststellen.
Dan nog iets over responsetijden .Op het transaktieschema zijn alle interakties aangegeven. Dialoogsimulatie wordt opgezet op basis van het transaktieschema. Per interaktie wordt de responsetijd op een waarde ingesteld die nog net akseptabel is voor de gebruiker. Deze ingestelde responsetijden kunnen worden overgenomen op het transaktieschema, bijvoorbeeld boven de pijlen naar links. Daarmee is een soort eigenschap van de transaktie vastgelegd. Hoe men met deze eisen moet omgaan, is aangegeven in de paragraaf Responsetijden. We kunnen de belangrijkste aspekten van een transaktie alsvolgt samenvatten:

- Menselijke handelingen en machinale verwerking worden samengevoegd tot voor gebruikers herkenbare eenheden: transakties.
- Op het transaktieschema worden beide genoemde aspekten in gebruikerstermen vastgelegd.
- Via dialoogsimulatie evalueert en akseptеert de gebruiker de geautomatiseerde manier van werken.
- De eisen ten aanzien van de responsetijden worden op een realistische manier vastgesteld.
- Via Transaktie analyse kan gekwantificeerd worden per transaktie en ontstaan per transaktie een aantal basisgegevens die van belang zijn voor gebruikers en automatiseerders.

In sommige systeemontwikkelingsmethoden komt het woord "transactions" voor. Meestal wordt het niet gedefinieerd, maar het blijkt bijna altijd iets te maken te hebben met de verwerking door de computer. Soms is een transaction de verwerking per interaktie. Dan gaat het dus om programmadelen tussen "READ SCREEN" en "DIS-

PLAY". Bij andere methoden bestaat er wel een relatie met het begrip transaktie, in die zin, dat een transaction de totale verwerking betreft van een transaktie. Dat is de rechterhelft van het transaktieschema. In die zin wordt het met name gebruikt wanneer er nog niet gekozen is tussen een batch- of on-line-toepassing. Wanneer die keus nog niet gemaakt is, valt er uiteraard nog niets te zeggen over menselijke handelingen tijdens de transaktie. Omdat in geen enkele systeemontwikkelingsmethode menselijke handelingen worden gekwantificeerd, is er ook geen behoefte aan het begrip transaktie. De gebruiker bestaat wel, maar alleen als lijdend voorwerp voor de informatieanalist, als geinterviewde, als geadresseerde van pakken systeemdokumentatie, als funktionaris die schermlay-outs goedkeurt. In deel 5 voor de transaktieanalist wordt aangegeven wat de kwantitatieve resultaten zijn van transaktie-ontwerp. Dan komt de gebruiker heel anders in beeld.
Het transaktie-ontwerp verloopt als volgt:
- Het maken van het transaktieschema`s.
- Het uitvoeren van dialoogsimulatie.
- Het maken van het definitieve transaktieschema.
- Het afdrukken van de schermlay-out en de transaktiedefinitie.
- Het uitvoeren van de ergonomische Transaktie analyse.
- Het vastleggen van de resultaten van Transaktie analyse.

Een en ander is weergegeven in Fig. 34.1. Wanneer om een of andere reden wordt afgezien van dialoogsimulatie, verloopt het transaktie-ontwerp als volgt:
- Het maken van het transaktieschema.
- Het uitvoeren van de ergonomische Transaktie analyse.
- Het vastleggen van de resultaten van Transaktie analyse.

Tenslotte kan men nog van een rudimentaire vorm van transaktieontwerp spreken, wanneer alleen een transaktieschema wordt ge-
Wanneer we deze drie vormen van transaktie-ontwerp met elkaar vergelijken blijkt de betrokkenheid van de gebruikers snel af te nemen. Daarnaast neemt echter ook de hoeveelheid attributen van het entiteitstype "Transaktie" af. Zo zijn alle kwantitatieve attributen bij de derde vorm van ontwerpen weggevallen. Bij de eerste vorm van ontwerpen verloopt de kommunikatie met de gebruikers optimaal en ontstaan alle attributen die nodig zijn om de ergonomische en sociale aspekten van een transaktie weer te geven. In het vervolg zullen we steeds uitgaan van de eerste manier van ontwerpen.
Transaktie-ontwerp is dus veel meer methodisch en omvattend dan het aloude "dialoogontwerp" van de meeste systeemontwikkelingsmethoden.

34.2 Waarom transaktie-ontwerp

Het woord "inspraak"geeft aan dat er sprake is van twee niveau's: het hoge niveau,dat in feite de gang van zaken bepaalt, en het lage dat mag proberen daar invloed op uit te oefenen. In veel bedrijven geeft het woord inspraak dan ook korrekt de verhouding aan tussen automatiseerders en gebruikers. Ten aanzien van de automatisering wordt de dienst uitgemaakt door automatiseerders. In veel systeemontwikkelingsmethoden blijkt de inbreng van de gebruikers niet noodzakelijk te zijn voor de voortgang van een projekt. In iedere methode is wel het punt te vinden waar de schermlay-outs aanwezig moeten zijn om verder te kunnen. In de beschrijving staat dan aangegeven dat ze in nauw overleg met de gebruiker moeten worden opgesteld en door hem moeten worden goedgekeurd. Maar of de gebruiker zich iets voor kan stellen bij schermlay-outs op A4-formaat is niet van belang voor de voortgang van het projekt. Of de dialoog aansluit bij de rest van de procedures, ligt niet vast. Zo gauw de verwerking door de computer vastligt, inclusief de besturing van dialoog, kan het technisch ontwerp worden gemaakt. Dat er binnen het ontwerp van een geautomatiseerd systeem niet een "werkeenheid" is, die past op de gebruikersorganisatie moet stemmen tot nadenken.
De automatiseerder denkt in schermen die elkaar op een of andere manier opvolgen en waar een hoeveelheid verwerking achter steekt. Met die verwerking is hij bezig in termen van programmafunkties en database calls. De gebruiker denkt aan het invoeren van orders. Hij vraagt zich af hoe vaak hij dat doet per dag en hoe het nu moet tijdens de pieken. Als een automatiseerder denkt aan het invoeren van een order, dan ziet hij dat er een proces wordt gestart, dat er bestanden worden geraadpleegd en gewijzigd. Hoe vaak dat per dag gebeurt is niet van belang voor zijn programmatuur. Dat heeft hoogstens iets te maken met de performance en daar bemoeien de specialisten zich wel mee.
Het is daarom van belang dat de entiteit "transaktie" wordt ingevoerd. De afbakening van een transaktie gebeurt dan ook door de gebruiker. Hij weet hoe de handmatige procedures beginnen en eindigen. Wanneer de transaktie is vastgelegd op een transaktieschema is een afgeronde, benoemde entiteit ontstaan. Die kan worden opgenomen in een dokumentatiesysteem, waaraan eigenschappen kunnen worden toegekend en waarvan gegevens kunnen worden berekend die van belang zijn voor het netwerkontwerp en voor de gebruiker. Gebruikers en automatiseerders werken nu met dezelfde eenheid. Gebruikers kunnen nu ook konkrete eisen stellen aan het ontwerp van transakties. De menselijke handelingen zullen immers door eindgebruikers worden uitgevoerd. Er zit enige flexibiliteit in de synchronisatie met andere ontwerpaktiviteiten. Het is bijvoor-

beeld niet zo dat het ontbreken van het transaktie-ontwerp het ontwerpen van het gegevensmodel tegenhoudt. Het transaktie-ontwerp loopt gedeeltelijk parallel met andere aktiviteiten. Maar onafhankelijk daarvan is voor de gebruikers heel nauwkeurig omschreven wat er van hen verwacht wordt en wat het betekent om mede-ontwerper te zijn.
Een aspekt dat zeker aan de orde moet komen is de benodigde tijd. Met name projektleiders interesseert dit onderwerp in hoge mate. Opvallend is wel dat "gebruikers interviewen" door iedereen als normale aktiviteit is geaksepteerd, maar minstens zo moeilijk in tijd is uit te drukken als transaktie-ontwerp. Ook de kommunikatie met een bepaalde gebruiker hangt af van de omstandigheden en de komplexiteit van het proces. Er is dus geen cijfer te geven voor de benodigde tijd voor transaktie-ontwerp. Wel staat een aantal zaken vast:
- Er is precies bekend wat er moet worden opgeleverd aan standaarddokumenten.
- De gebruikers hebben de geautomatiseerde procedure aan den lijve ondervonden door dialoogsimulatie.
- Er zijn nauwkeurige eisen gesteld aan de performance.
- Na enige ervaring in het ontwerpen van transakties zullen informatie-analisten de benodigde tijd nauwkeuriger kunnen schatten dan bij interviews.

De beschikbaarheid van transaktieschema's maakt de uitvoering van Transaktie analyse mogelijk. Dan moet ook de verwerking door de computer gekwantificeerd worden. Natuurlijk is er dan alleen nog maar een logisch gegevensontwerp beschikbaar, maar toch is het op dat moment al verhelderend om de verwerking te kwantificeren in aantallen logische databasecalls. Het rekenmodel van Transaktie analyse maakt het mogelijk om heel snel worst cases door te rekenen. Daardoor wordt voorkomen dat verwerkingsprocessen worden ontworpen die nooit kunnen voldoen aan de responsetijdeisen. In vele projekten had dat alleen al veel ellende achteraf bespaard. Het gaat dus niet om de berekening van responsetijden, maar om het tijdig vaststellen van de onhaalbaarheid. Dit aspekt is behandeld in de paragraaf Vijf soorten vakmanschap. Wanneer Transaktie analyse is uitgevoerd kunnen de tijdsbestedingsoverzichten worden gemaakt en dan worden de sociale aspekten bespreekbaar. Hoe de synchronisatie met het gegevensontwerp en het funktionele ontwerp ook wordt gerealiseerd, voor het eind van het logisch ontwerp moet het transaktie-ontwerp volledig zijn uitgevoerd. Zowel het automatiseringsmanagement als het gebruikersmanagement is daarvoor verantwoordelijk.
Transaktie-ontwerp heeft ook te maken met de lokatie van gegevens: wanneer transakties zijn ontworpen kunnen ze gekoppeld worden aan werkplekken. Werkplekken zijn geografisch bepaald, per

transaktie is bekend welke gegevens gebruikt worden, de geografie van het gebruik van gegevens ligt nu dus vast, en er kan een begin gemaakt worden met het ontwerp van de gegevensdistributie. Via de resultaten van Transaktie analyse kunnen nu de konsekwenties van bepaalde situaties worden berekend. Daarmee wordt dan de eerste, konkrete, aanzet gegeven tot het netwerkontwerp.
Transaktie-ontwerp betekent de akseptatietest tijdens het logisch ontwerp! Daarom kunnen projekten in eerste instantie worden uitgevoerd tot het logisch ontwerp, waarna beslissingen genomen kunnen worden ten aanzien van aantal terminals, konfiguratie, etc.

34.3 Transaktie-ontwerp en Transaktie analyse

De gegevens voor het rekenproces van Transaktie analyse en het rekenproces zelf zijn altijd hetzelfde. De invoer is altijd een transaktiedetailschema. Dit detailschema is rechtstreeks afgeleid van het transaktieschema. Het transaktieschema kan echter, afhankelijk van de beschikbare informatie, meer of minder nauwkeurig zijn. Zo kan het gebeuren dat tijdens een vooronderzoek nog geen transaktie-ontwerp met gebruikers heeft plaatsgevonden, maar dat de automatiseerders toch wel een voorlopig idee hebben van transakties die op een bepaalde werkplek moeten gebeuren. Zo zal er bij een kassa in een supermarkt bijna zeker worden afgerekend. Wanneer het computersysteem later een voorstel moet doen voor een inkooporder, dan is het kontroleren van het voorstel via een beeldscherm best vast te leggen op een transaktieschema. Dat zou er uit kunnen zien zo als in Fig. 34.2 is aangegeven.
Uit het voorbeeld blijkt dat er nog geen schermlay-out bekend is. Het gaat om het aantal schermen met een gemiddeld aantal tekens. Dit is dus geen transaktie-ontwerp zoals dat in dit hoofdstuk is gedefinieerd. Het is het maken van een transaktieschema, met of zonder de gebruiker, om de automatiseerders in een vroeg stadium enige cijfers ter beschikking te stellen.
Op basis van dit transaktieschema wordt een detailschema gemaakt. Aangezien lang niet alle resultaten van belang zijn, hoeven niet alle parameters in het detailschema te worden opgenomen. Met name de technische parameters, die te maken hebben met de soort terminal en het verkeer, hoeven niet te worden ingevuld. We noemen dit de globale Transaktie analyse . In de praktijk is gebleken, dat hier interessante gegevens uit naar voren kunnen komen. Zo bleek bijvoorbeeld eens dat de tijd die sommige funktionarissen nodig zouden hebben om hun transakties iedere dag uit te voeren, de acht uur zou overschrijden. In een ander projekt ontstond een goed inzicht in het aantal printers dat nodig was om een aantal werkplekken van voldoende printkapaciteit te voorzien. Dat leidde weer tot duidelijke performance-eisen voor het computersysteem en

het netwerk: interaktieve toepassingen gekombineerd met remote printers!
De volgende vorm van Transaktie analyse is de logische Transaktie analyse . We zitten dan midden in het logisch ontwerp: het transaktie-ontwerp moet af zijn. Er kan nu een nauwkeurig detailschema gemaakt worden, op een veld nauwkeurig, aangezien de schermlay-out is gegeven. De verwerking door de computer kan nog niet nauwkeurig worden aangegeven, want de database of de bestanden zijn immers pas op logisch niveau gedefinieerd. De verwerking moet daarom ook in logische calls op het detailschema tot uitdrukking worden gebracht. Terecht wordt in (7), VOL II, pag. 366, gezegd: "This response time may be evaluated in terms of the end users absolute response time (performance) requirement. Estimated transaction responsetimes which appear to exceed the end users maximum acceptable response time so indicate potential response-critical transactions".
Met andere woorden, hoewel het gaat om het logisch ontwerp, is het van belang nu reeds te letten op responsetijden. Niet om die te berekenen, maar om te zien of voor de kritische interakties aan de eisen kan worden voldaan. Ervaren informatie-analisten moeten daar na Transaktie analyse een mening over hebben. Wanneer blijkt dat de responsetijden lang niet gehaald kunnen worden, kan men nu al beginnen aan de eerste iteratie: terugkoppeling naar de gebruiker. Het resultaat van die terugkoppeling is ofwel een ander transaktie-ontwerp ofwel aangepaste responsetijden die weer via dialoogsimulatie hard zijn gemaakt.
Wanneer tijdens het logisch ontwerp gegevens nodig zijn over hoeveelheden verkeer, dan kunnen de daarmee verband houdende parameters in de detailschema's worden verwerkt. In principe doen zich twee situaties voor.
Het gaat om een projekt op bestaande hardware. Dan zijn de gegevens van de terminals bekend en kan het detailschema meteen korrekt worden ingevuld.
Het kan ook zijn dat de hardware nog gekozen moet worden. Dan moeten er een of twee situaties gekozen worden. Is er absoluut geen inzicht in de uiteindelijke keus uit hardware, dan kan men het beste van een 3270-achtige terminal uitgaan. Veel in blockmode werkende terminals hebben een vergelijkbaar screenmanagement. Als worst case kan daarnaast een domme teletype-achtige terminal worden genomen. Het invoeren van een ander terminaltype betekent, dat per detailschema twee soorten parameters moeten worden aangepast op alle plaatsen waar ze voorkomen. Tenslotte komen we bij de technische Transaktie analyse . Wanneer geen globale en geen logische Transaktie analyse is uitgevoerd zijn alleen de transaktieschema's van het transaktie-ontwerp aanwezig. Als wel een van beide analyses is uitgevoerd zijn er detailschema's aanwezig. Die

TRANSAKTIESCHEMA centraal

Transaktienaam: Registratie ontvangen goederen

Menselijke handelingen en bewerkingen	Transport	Machinale verwerking
Intoetsen van goederen-ontvangst-identifikatie van pakbon		
	----)	Opzoeken order, orderregels formeren van n schermen. Display scherm
	(----	
Lezen interpreteren en vergelijken. Bij afwijkingen: intoetsen van korrekte aantal, in enkele gevallen inclusief artikelnummer (niet bestelde goederen)	----)	Verwerking, eventueel volgende scherm displayen
	(----	
Idem	----)	Idem, na laatste scherm, bijwerken bestanden displayen "OK".
	(----	
Lezen "OK", volgende pakbon of terug naar menuscherm		
	----)	Naar begin van transaktie of naar menuscherm.

Fig. 34.2 Voorbeeld van een globaal transaktieschema.

detailschema`s hoeven nu alleen te worden aangevuld met die gegevens die tijdens het technisch ontwerp bekend zijn geworden. Meestal gaat het om het fysieke database-ontwerp en het soort terminals.
Ten aanzien van responsetijden moet de situatie nu duidelijk zijn. Niet op 10% nauwkeurig, maar wel op 100% of 200%. Wanneer daarover tijdens het technisch ontwerp wordt gesproken, zijn we verder dan in de meeste praktijksituaties, waar deze gesprekken meestal plaatsvinden na de invoering. Gesprekspartners zijn hier: informatie-analisten, technische ontwerpers, systeemspecialisten, transaktie-analisten en datakommunikatiespecialisten.
Op deze manier is een transaktie een ontwerpgegeven, dat zijn leven begint tijdens het logisch ontwerp en soms eerder, en dat via Transaktie analyse cijfermateriaal oplevert voor responsetijden en netwerkontwerp.

34.4 Transaktie-ontwerp en andere ontwerpaktiviteiten

In het algemeen kan gezegd worden dat elke systeemontwikkeling uiteindelijk leidt tot databases of bestanden en programma`s. In welke volgorde en in welke onderlinge relatie beide ontworpen worden is in dit verband niet van belang. Informatie-analisten en systeemontwerpers zijn goed in staat om zonder inbreng van gebruikers informatieverwerkende systemen te bouwen. Het bestaande bedrijf wordt in kaart gebracht op basis van gegevens en processen. Vervolgens worden gegevensmodellen en procesmodellen ontworpen, uiteindelijk resulterend in databases of bestanden en programma`s.
Wanneer transaktie-ontwerp wordt geaksepteerd als de beste manier om gebruikers te betrekken bij het ontwerp en cijfermateriaal te verzamelen voor het netwerkontwerp , moet deze methode worden geintegreerd in de overige aktiviteiten tijdens het logisch ontwerp. Op het transaktieschema is in de rechterkolom, in gebruikerstermen per interaktie de verwerking beschreven en de daarbij benodigde gegevens. Het rechter deel is immers het machinedeel van de transaktie. Per transaktie moet nu van iedere interaktie worden vastgesteld welke verwerkingsfunktie daar wordt beschreven en welke bestanden, records en items daarbij gebruikt worden. De verwerking moet worden vergeleken met de funktiemodellen of algoritmen die ontworpen zijn. Van de bestanden, records of items moet gekontroleerd worden of ze in het gegevensmodel voorkomen.
Wanneer systeemontwerpers het systeem ontwerpen op basis van de uitgevoerde analyses wordt in principe de bestaande situatie geautomatiseerd. Transaktie-ontwerp biedt de gebruiker echter de mogelijkheid om kreatief om te gaan met de automatisering. Dat is een wezenlijk sociaal aspekt van deze aanpak. Het kan dus zijn

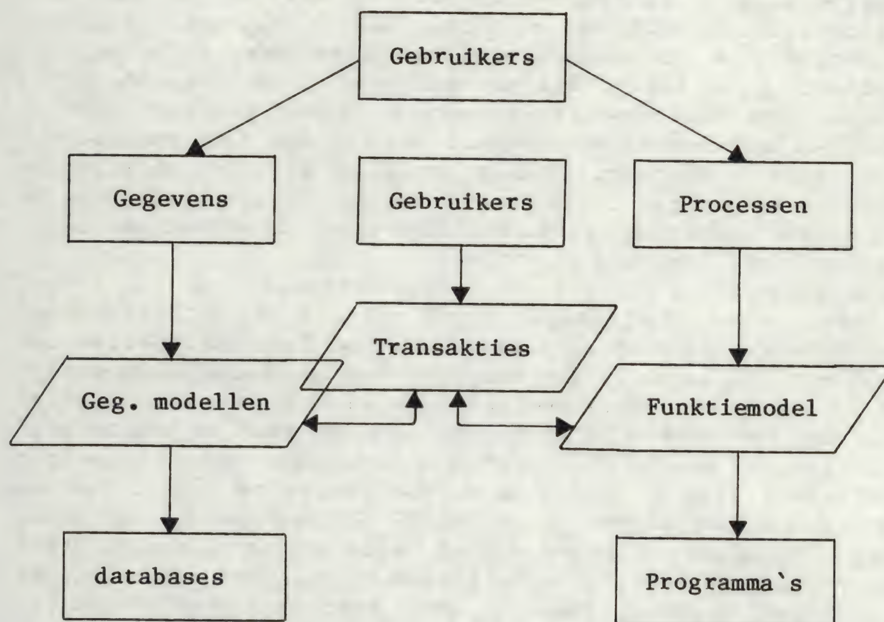


Fig. 34.3 Gebruikers tijdens analyse en ontwerp

dat de gebruiker met een aantal voorstellen komt voor niet geanalyseerde procedures. Zo zou iemand van een verkoopafdeling op het idee kunnen komen om klanten die hij aan de telefoon krijgt, informatie te verstrekken over de situatie van lopende orders, die in de handmatige situatie niet gegeven kon worden. Deze nieuwe transakties kunnen van grote invloed zijn op het hele ontwerp. Daarom moet de rechter kolom van het transaktieschema nauwkeurig worden vergeleken met de reeds ontworpen modellen.
Dit proces zou geautomatiseerd kunnen worden door het transaktieschema te vertalen naar een automatiseringsdokument, waarop de gebruikersterminologie voor de verwerking is vervangen door gegevens- en procesnamen. Beide kunnen dan vergeleken worden met een data dictionary. Kortom, het transaktie-ontwerp dient niet als vervanging voor enige projektaktiviteit, maar moet afgestemd worden op gegevens- en funktie-ontwerp.

34.5 Transaktie-ontwerp en projektaanpak

Automatiseringsprojekten kunnen op allerlei manieren worden opgezet. Bij kleine bedrijven wordt soms de hele automatisering als een projekt beschouwd. Bij grote bedrijven, die meestal reeds ge-

deeltelijk zijn geautomatiseerd, betreft een projekt vaak een subsysteem. Daar heeft men op basis van de opgedane ervaring, vaak gekozen voor een systeemontwikkelingsmethode, om zo de hele automatisering te kunnen blijven beheren. In de volgende paragraaf gaat het om transaktie-ontwerp in omgevingen waar gekozen is voor een systeemontwikkelingsmethode, al dan niet ondersteund door een data dictionary. In deze paragraaf gaat het om transaktie-ontwerp in diverse fasen, zoals men die in elk projekt op de een of andere manier aantreft. Per fase wordt de werkwijze besproken en toegelicht met een figuur.

- Transaktie-ontwerp tijdens het vooronderzoek.

Deze fase in de projektaanpak wordt ook wel definitiestudie of toepasbaarheidsonderzoek genoemd. Hoe deze fase inhoudelijk ook gedefinieerd is, bijna altijd komt het begrip kosten/baten-analyse er in voor. De bedrijfsleiding wil in een zo vroeg mogelijk stadium inzicht hebben in de kosten. Als het gaat om een bedrijf dat, gezien de geografische situatie, te maken zal krijgen met een netwerk tussen de vestigingen, dan kunnen de kosten daarvan een grote invloed hebben op de GO/NO GO beslissing. Het gaat daarbij niet alleen om de bouwkosten, maar ook om de exploitatiekosten van het netwerk. De schattingen tijdens het vooronderzoek zijn uiteraard globaal, want het gaat meer om kostenindikaties. In de formules die gebruikt worden, komen de netwerkkosten niet voor. Dat kan ook niet, omdat nog niet bekend is welke vestigingen worden uitgerust met een computer, welke transakties per vestiging zullen worden uitgevoerd, welke hardware daarvoor gebruikt zal worden enzovoort. Tijdens het vooronderzoek is de informatieanalist met heel andere zaken bezig: de organisatie wordt in kaart gebracht, evenals de bestaande methoden en procedures, de knelpunten, de hoofdfunkties van het systeem. Bij grote bedrijven zullen zijn gesprekspartners de managers zijn, bij bedrijven met een ondiepe organisatiestruktuur komen de werkplekken van de eindgebruikers in beeld. In welke fase het projekt verkeert is voor de eindgebruiker in een bepaald opzicht niet van belang. Wanneer hij bijvoorbeeld via een tweedaagse kursus is ingewijd in het beeldschermgebeuren, kan hij op elk willekeurig moment een bijdrage leveren tot het ontwerpen van transakties voor zijn werkplek. Dat geldt nog sterker voor de kwantitatieve gegevens van zijn werk. Het aantal orders dat hij per dag verwerkt weet hij op elk willekeurig moment, onafhankelijk van het feit of er geautomatiseerd wordt of niet. Dat betekent dat men tijdens het vooronderzoek best kan beginnen met het transaktie-ontwerp. Dan worden er dus transaktieschema's gemaakt en dialoogsimulatie wordt uitgevoerd. Tijdens het vooronderzoek evalueren eindgebruikers het werken met beeldschermen. Het blijkt te werken.

Op basis van de besproken transaktieschema's kan de logische

Transaktie analyse worden uitgevoerd zoals is beschreven in de paragraaf Transaktie-ontwerp en Transaktie analyse. Daarmee is dan het gestelde doel bereikt: een globaal netwerkontwerp en een goede indikatie van de kosten. Dit wordt verder uitgewerkt in het hoofdstuk Netwerkontwerp.
Natuurlijk is het vooronderzoek niet de meest logische fase om transakties te ontwerpen. Er is alleen aangegeven dat het mogelijk is. Wanneer de bedrijfsleiding in de kosten/baten-analyse het netwerk mee wil nemen, dan kan dat: het vooronderzoek gaat daardoor langer duren, maar het ontwerpen van transakties gaat sneller dan het ontwerpen van gegevensmodellen of procesmodellen. Verder is de tijd die er ingestoken wordt niet verloren, want wanneer het projekt doorgaat, is een deel van het logisch ontwerp gereed. Transaktieschema`s, beeldschermlay-outs en kwantitatieve gegevens zijn beschikbaar. De nauwkeurigheid van die gegevens hangt af van de uitgevoerde Transaktie analyse: de globale, de logische of de technische. Eigenlijk gaat het dus om een verschuiving van aktiviteiten als de nauwkeurigheid van de kosten-/baten-analyse dat vereist. Zonder transaktie-ontwerp was de verschuiving helemaal niet mogelijk.
- Transaktie-ontwerp tijdens de analysefase.
Voorafgaand aan het logisch of funktioneel ontwerp wordt de bestaande situatie geanalyseerd. Bij sommige systeemontwikkelingsmethoden maakt deze objektanalyse deel uit van het logisch ontwerp. Tijdens de analyse worden bestaande processen en gegevens in kaart gebracht. De informatie-analist komt daarbij zeker terecht op werkplekniveau. Niet omdat de lokatie van de werkplek van belang is, of omdat de persoon op een bepaalde werkplek hem interesseert, maar doordat er gegevens verwerkt worden, berekeningen of kontroles worden uitgevoerd. Via interviews en dokumenten moet hij zich een beeld vormen van die aktiviteiten. Wanneer de gebruiker is voorbereid op het werken met beeldschermen of reeds enige ervaring heeft, kan de informatie analist overwegen samen met hem de processen in kaart te brengen door ze te vertalen naar de beeldschermsituatie. Dan lopen analyse en ontwerp in elkaar over. Het is de vraag of het verstandig is om te ontwerpen tijdens de analysefase. Er zijn echter omstandigheden waarin de kommunikatie met de gebruikers stroef en moeizaam verloopt. De bereidheid om gegevens te verstrekken hangt dan vaak niet alleen af van de kennis van de gebruiker: soms speelt tegenzin tegen de hele automatisering of onzekerheid over de inhoud van de toekomstige funktie een veel grotere rol. In zulke situaties kan dialoogsimulatie wonderen verrichten. Kortom, de analysefase is niet het aangewezen moment voor transaktie-ontwerp, maar er kunnen zich situaties voordoen waarbij de voordelen opwegen tegen de nadelen.

- Transaktie-ontwerp tijdens het logisch ontwerp.
Dit is het moment waarop het transaktie-ontwerp optimaal funktioneert. De bestaande situatie is in kaart gebracht, bijvoorbeeld in de vorm van funktie- en informatiemodellen. Nu worden de proces- en gegevensmodellen ontworpen. Dat betekent dat de verwerkingsfunkties en de gegevens zoals aangegeven op de transaktieschema`s kunnen worden vergeleken met de ontworpen modellen. Daarnaast vormen alle transaktieschema`s tezamen een goede basis voor de dialoogstruktuurdiagrammen. Er zijn talloze bedrijven waarvan de gebruikers nooit in staat zullen zijn om proces- en gegevensmodellen te lezen, laat staan te beoordelen. Het enige wat ze begrepen hebben is dat de computer een deel van hun werk gaat overnemen en dat ze met een beeldscherm gaan werken. Voor deze gebruikers is het transaktie-ontwerp het enige konkrete: het transaktieschema is geschreven in hun taal, dialoogsimulatie hebben ze uitgevoerd. Rekenprocessen kunnen ze meestal precies omschrijven, en achteraf, tijdens de invoering, gemakkelijk kontroleren. Hoe de gegevens in een computer worden opgeslagen is voor gebruikers niet van belang. In welke taal ze ook beschreven zijn, modellen zijn er voor de automatiseerders. De kommunkatie met de gebruiker over het te ontwerpen informatiesysteem loopt wat het interaktieve deel betreft via transaktie-ontwerp. Zoals is aangegeven in diverse paragrafen over responsetijden is het van belang aan het eind van het logisch ontwerp na te gaan of de gespecificeerde responsetijden gehaald kunnen worden. Er is nu bekend hoe het gegevensmodel in elkaar zit. Aan de hand van de gewenste verwerkingsfunkties op de transaktieschema`s en de bijbehorende eisen kan worden vastgesteld of er situaties zijn waarvan men nu al kan vaststellen of er responsetijdeisen zijn die zeker niet haalbaar zijn. Terugkoppeling naar de gebruiker is dan noodzakelijk: hij moet nu beslissen of hij een ander transaktieontwerp zal voorstellen of de responsetijdeisen zal aanpassen. Met dialoogsimulatie is het eenvoudig de nieuwe tijden hard te maken en te laten goedkeuren door verscheidene gebruikers.
- Transaktie-ontwerp tijdens het technisch ontwerp.
In principe vindt het transaktie-ontwerp plaats tijdens het logisch ontwerp. Er zijn minstens twee situaties waarin ook tijdens het technisch ontwerp nog iets aan het transaktie-ontwerp gedaan moet worden. De eerste situatie ontstaat wanneer tijdens het technisch ontwerp hardware wordt gekozen die veel meer of veel minder kan dan men tijdens het logisch ontwerp had aangenomen. Bij veel meer kan men bijvoorbeeld denken aan kleuren, scrolling, het aantal funktietoetsen, hardcopy, spraak, gegevens uit andere computersystemen, bij veel minder aan het aantal funktietoetsen, performance in verband met responsetijden, het aantal schermen en/of printers. Het kan betekenen dat ontworpen transakties ver-

kan transaktie-ontwerp net zo zinvol zijn als bij een projekt waarin niet wordt geprototyped. De inbreng van gebruikers via transaktieschema's en dialoogsimulatie is een prima start van prototyping. Als het aksent ligt bij het simuleren van processen met behulp van talen als APL en SIMULA, dan is transaktie-ontwerp bijna noodzakelijk. Tenslotte kan worden opgemerkt dat prototyping op zich geen relatie heeft met het netwerkontwerp. Bij het ontwerpen van netwerken is weleens sprake van simuleren, maar dat is iets heel anders dan prototyping tijdens het logisch ontwerp. Als logisch ontwerp in verband moet worden gebracht met netwerkontwerp via Transaktie analyse, dan moeten er in ieder geval transaktieschema's worden gemaakt, onafhankelijk van de ontwerpmethode. Wanneer geen dialoogsimulatie wordt toegepast, omdat men wil prototypen, kunnen ze het best zo laat mogelijk in het logisch ontwerp worden opgesteld. Dan zijn gebruikers al betrokken geweest bij het ontwerp en kan er gesproken worden van uitgekristalliseerde transakties.
In de twee delen voor de informatie-analist en de transaktie-analist wordt verder uitgewerkt hoe de procedures verlopen in diverse omgevingen.

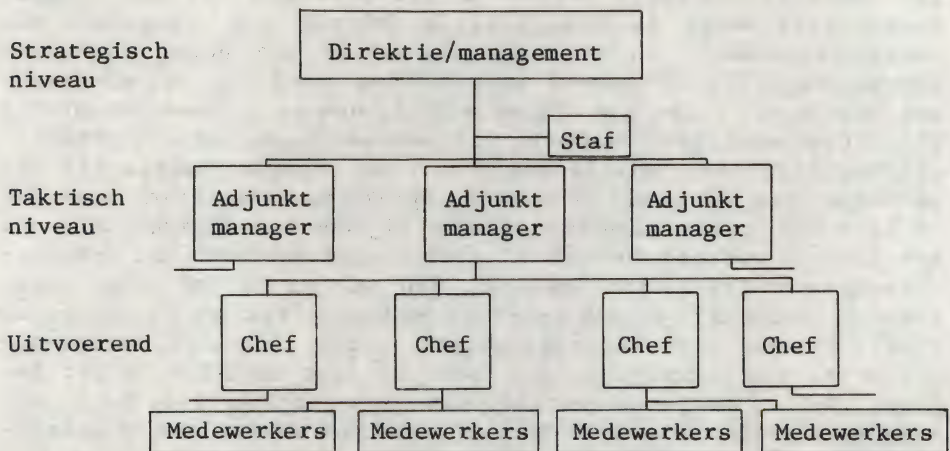
34.7 Transaktie-ontwerp in het grote geheel

Alsof automatiseren en zich nog niet ingewikkeld genoeg is komen

vallen of aangepast moeten worden en soms nieuwe transakties moeten worden ontworpen. Deze iteraties kunnen beter plaats hebben tijdens het technische ontwerp dan bij de invoering.
De tweede situatie kan zijn: de systeemontwerper, systeemspecialist en transaksaktie-analist komen tot de konklusie dat sommige responsetijden niet haalbaar zullen zijn. We gaan ervan uit dat alle mogelijkheden voor het ontwerpen van een fysieke database zijn overwogen. Terugkoppeling naar de gebruiker heeft plaats op dezelfde manier als bij het logisch ontwerp. Wanneer dat proces leidt tot gewijzigde of nieuwe transakties moet uiteraard voor een deel het logisch ontwerp worden herzien. De nieuwe transaktieschema's vormen daarvoor de startdokumenten.

34.6 Transaktie-ontwerp en systeemontwikkelingsmethoden

Zoals bekend is, bestaat er een groot aantal systeemontwikkelingsmethoden. In (9) worden van de belangrijkste de hoofdlijnen aangegeven. In (6) worden er een aantal met elkaar vergeleken. Bij de ene methode blijkt het aksent te liggen op het informatieplan, bij de andere op de systeemdokumentatie, bij weer een andere op het logisch ontwerp. Bij de meeste methoden is de gebruiker



Funktionaris: Medewerker, manager, chef, direktielid. (13)

Fig 34.4 Een organisatie

vast om welke funktionarissen het gaat. Een funktionaris kan zowel een uitvoerende taak als een managementtaak hebben. Uiteraard gaat het alleen om de mensen die betrokken zijn bij de automatiring omdat ze een beeldscherm gaan gebruiken.

- De geografische situatie.

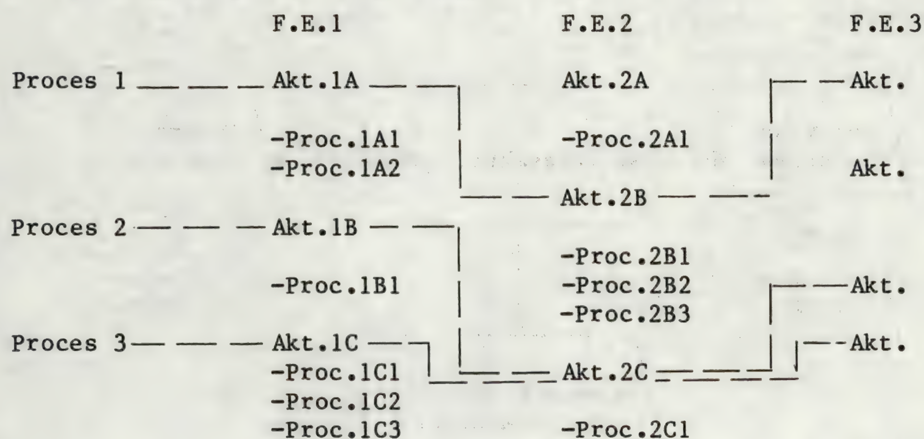
In verband met het ontwerpen van netwerken is de geografische situatie van belang. Daarom wordt de geografische situatie in kaart gebracht, zoals is aangegeven in Fig. 34.5. In de tabel is de meest uitgebreide situatie weergegeven.

- Een concern heeft kantoren in een aantal plaatsen die via interlokale lijnen met elkaar kunnen worden verbonden.
- Per vestigingsplaats kunnen verscheidene gebouwen voorkomen die via lokale lijnen met elkaar zijn verbonden.
- Per gebouw kan een local area network beschikbaar zijn.
- Per verdieping of afdeling of groep gebruikers kan een clustercontroller met een aantal beeldschermen zijn opgesteld. Daarbij kan het best zijn dat er niet een clustercontroller, maar een afdelingscomputer is geplaatst.
- Op de werkplek staat in ieder geval een beeldscherm.

Het soort netwerk is dus alleen bedoeld om aan te geven hoe in het meest ingewikkelde geval het netwerkverkeer kan zijn opgebouwd. Noodzakelijk is dat natuurlijk niet: het kan best zijn dat

Geografische eenheid	Soort netwerk
Bedrijf	Wide area network
Vestigingsplaats	Netwerk met lokale lijnen
Vestiging, gebouw	Local area network
Verdieping, afdeling	Cluster
Werkplek	Beeldscherm

Fig 34.5 Geografie



F.E. = Funktionele eenheid
Proc.= Procedure
Akt. = Aktiviteit

Fig 34.6 Funktionele struktuur

een cluster via een wide area network verbonden is met het mainframe.
De splitsing van een vestiging in verdiepingen of afdelingen is daarmee ter diskussie gesteld en afhankelijk van de konkrete situatie. In het kader van deze paragraaf gaat het evenals bij de organisatie om het laagste niveau: de werkplek. Iedere werkplek is dus geografisch bepaald, maximaal op basis van plaatsaanduiding, verdieping, gebouw, vestigingsplaats. De lokaties van de werkplekken zijn de basis voor de topologie van het netwerk.
- De funktionele indeling.
In elk bedrijf zijn een aantal funktionele eenheden aan te wijzen. Het aantal en de soorten zijn afhankelijk van de bedrijfsdoelstelling. Bij een produktiebedrijf zal de situatie anders zijn dan bij een verkoopkantoor. Per funktionele eenheid kunnen we een aantal aktiviteiten aanwijzen, die weer uiteenvallen in procedures. Evenals bij de vorige twee indelingen gaat het om het laagste niveau. De aktiviteiten moeten zover afgebroken worden in procedures dat er een relatie ontstaat met de funktionaris die ze uitvoert. Een funktionaris voert een of meer procedures uit. Wanneer dus een procedure wordt uitgevoerd door een groep funktionarissen, is de detaillering onvoldoende. Natuurlijk kunnen de medewerkers van een groep best allemaal dezelfde procedures uitvoeren. Per funktionaris ligt dan immers vast welke procedure hij uitvoert.

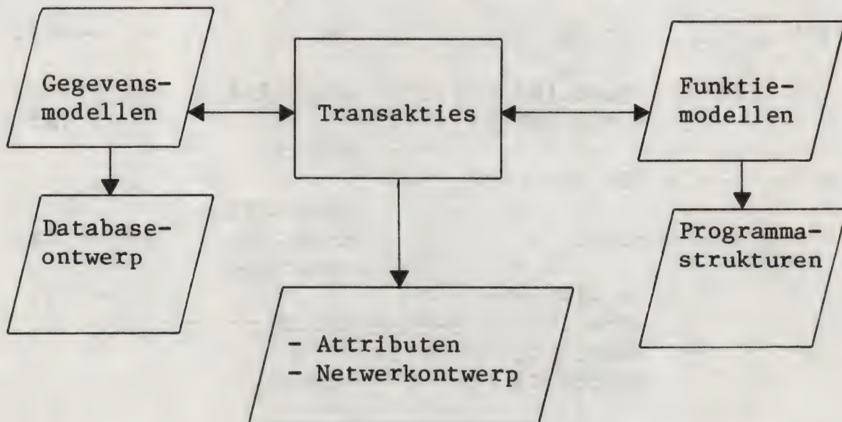


Fig 34.7 Automatisering

- De automatisering.
Evenals de rest van het bedrijfsgebeuren kan ook de automatisering een komplex geheel zijn van aktiviteiten, methoden, fasen, hardware en softwarebeperkingen, vakmensen, kommunikatieproblemen. In de analysefase worden bestaande gegevens en processen in kaart gebracht. Er worden gegevensmodellen en funktiemodellen gemaakt en eventueel gekoppeld aan een data dictionary. Dan volgt het moeilijkste: de overgang van analyse naar funktioneel ontwerp. Er ontstaan programmastrukturen, logische databases, schermlayouts, beschrijvingen. Wanneer tijdens het logisch ontwerp transakties worden ontworpen als aangegeven in het hoofdstuk Transaktie-ontwerp, onstaat de mogelijkheid om de vier terreinen aan elkaar te koppelen. We hebben gezien dat de struktuur van de vier genoemde terreinen zeer komplex kan zijn. Onafhankelijk daarvan, hebben we de laagste niveaus vastgesteld: funktionaris, werkplek, procedure en transaktie.
- De funktionaris of medewerker is de man met een bepaalde taak/ funktie-omschrijving. Dat is de werknemer zoals hij bij personeelszaken bekend is en zoals hij zijn taak verricht binnen het bedrijf.
- De werkplek is zowel de plaats waar de handmatige situatie zich afspeelt als de plek waar in de toekomst een beeldscherm wordt opgesteld. De werkplek is geografisch bepaald. Het gebruik van gegevens is daarmee eveneens geografisch bepaald. Kommunikatie met andere werkplekken betekent, in de geautomatiseerde situatie, een verbinding met andere werkplekken.
- De procedure is een geheel van handelingen dat de funktionaris uitvoert ter verwerking van gegevens. De detaillering moet zover gaan dat de informatie-analist zich gemakkelijk een transaktie kan voorstellen als vervanger van de procedure. In deel 4 wordt dit verder uitgewerkt in de paragraaf Van analyse naar transakties
- De transaktie is de beeldschermversie van een procedure. In vorige paragrafen is behandeld hoe transakties ontstaan en wat ermee gebeurt. Dankzij methoden als dialoogsimulatie is een transaktie tevens "eigendom" van de funktionaris. De koppeling kan nu een aantal belangrijke, konkrete resultaten opleveren.
- In de analysefase worden procedures in kaart gebracht waarbij gegevensmodellen en funktiemodellen ontstaan. Tijdens de analyse moet ook de geografische situatie in kaart worden gebracht, zodat procedures gekoppeld kunnen worden aan lokaties binnen het bedrijf. Tenslotte moet worden vastgelegd welke funktionaris bepaalde procedures uitvoert.
- In de bestaande taak/funktie-omschrijving moet zijn vastgelegd welke procedures bij een funktionaris thuis horen. Met andere woorden, de taak/funktie-omschrijving zou een kontrolemiddel moe-

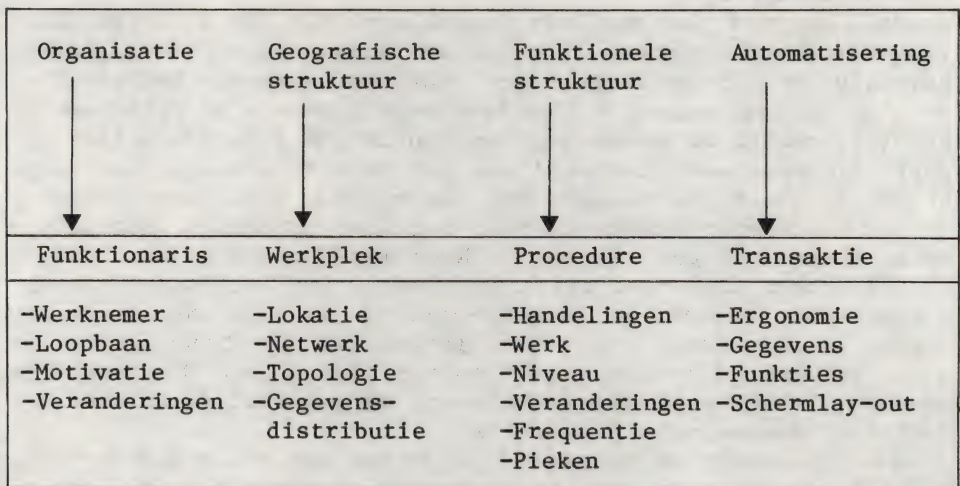


Fig. 34.8 De koppeling

ten zijn voor de informatie-analist. Een taak/funktie-omschrijving geeft de sociale aspekten weer van de huidige situatie. Wanneer men konkreet wil zijn over de sociale aspekten van de automatisering, moet er wel zoiets beschikbaar zijn als taak/funktie-omschrijvingen. In ieder geval zullen de huidige procedures op werkplekniveau beschreven moeten zijn. Niet in de modellen van de
- In de bestaande taak/funktie-omschrijving moet zijn vastgelegd welke procedures bij een funktionaris thuis horen. Met andere woorden, de taak/funktie-omschrijving zou een kontrolemiddel moeten zijn voor de informatie-analist. Een taak/funktie-omschrijving geeft de sociale aspekten weer van de huidige situatie. Wanneer men konkreet wil zijn over de sociale aspekten van de automatisering, moet er wel zoiets beschikbaar zijn als taak/funktie-omschrijvingen. In ieder geval zullen de huidige procedures op werkplekniveau beschreven moeten zijn. Niet in de modellen van de automatiseerders maar in dokumenten van personeelszaken .
- De nieuwe taak/funktie-omschrijving kan nu gemaakt worden. Aan het eind van het logisch ontwerp is bekend welke procedures worden vervangen door transakties. Aangezien transaktie-ontwerp inhoudt dat de gebruiker op de simulator al gewerkt heeft met de transakties, kan de funktionaris ook heel konkreet meepraten over de nieuwe inhoud van zijn funktie. Dankzij de ergonomische resultaten van Transaktie analyse is de funktie ook kwantitatief be-

kend: aantal uren achter het beeldscherm, aantal uren per transaktie, aantal transakties om de nieuwe funktie uit te voeren.
- Het gebruik van gegevens is nu geografisch bepaald. Per transaktie is bekend welke gegevens worden gebruikt. Een transaktie vervangt een of meer procedures en procedures zijn geografisch bepaald door de werkplek waar ze worden uitgevoerd. Daarmee ligt vast op welke lokatie gegevens worden gebruikt en is tevens de basis gelegd voor de gegevensdistributie.
- We kunnen een netwerk omschrijven als het middel om het verschil tussen gegevenslokatie en gegevensgebruik goed te maken. Na het transaktie-ontwerp is bekend op welke lokaties welke gegevens worden gebruikt. De lokatie van gegevens kan bepaald zijn, onafhankelijk van het gebruik, maar soms ligt de lokatie vast en moet een netwerk ontworpen worden om het gebruik mogelijk te maken. Soms wil men de lokatie laten afhangen van eisen die aan het netwerk worden gesteld. In die gevallen moet een aantal alternatieven worden berekend.
- De organisatorische konsekwenties kunnen nu tijdens het logisch ontwerp bottom-up worden vastgesteld. De nieuwe taak/funktie-omschrijvingen vormen daarvoor de basis. Natuurlijk kan men tijdens het vooronderzoek of de definitiestudie reeds nadenken over sociale en organisatorische gevolgen van automatisering. In deze fase worden vaak organisatie-adviseurs ingezet. Toch kan het dan alleen gaan over de grote lijnen. Enerzijds omdat men het detailniveau van transakties nog niet bereikt heeft, anderzijds omdat in dat stadium de gevolgen van de automatisering nog niet besproken kunnen worden met gebruikers die nog niet kunnen overzien wat de automatisering voor hun funktie inhoudt. Wanneer het transaktie-ontwerp is uitgevoerd is dat wel het geval. Dan begrijpt de gebruiker ook welke procedures vervallen, welke procedures vervangen worden door transakties en welke nieuwe transakties ontworpen zijn. In die situatie kunnen de organisatorische gevolgen zelfs vaak door de gebruikers zelf worden vastgesteld.
Transaktie-ontwerp levert dus, gezien het bovenstaande bijdragen in het inzicht in:
- Veranderingen voor de gebruikers
- Nieuwe taak/funktie-omschrijvingen
- Gegevensdistributie
- Netwerkontwerp.

34.8 Benodigde tijd

Transaktie-ontwerp kan gezien worden als de invulling van twee witte vlekken in systeemontwikkelingsmethoden. De ene witte vlek betreft de methodische kommunikatie tussen gebruikers en automatiseerders, de tweede de kommunikatie tussen systeemontwerpers en

netwerkontwerpers. De invulling van die twee vlekken kost tijd. Iedereen is geneigd dat te beschouwen als extra tijd. Als echter konsekwent per projekt een evaluatie zou plaats vinden, zou blijken dat de invoering, de kommunikatie met gebruikers, het aanpassen van de toepassing een hoeveelheid extra tijd heeft gekost, waar niemand nog graag over praat. Dat is waarschijnlijk ook de reden dat evaluaties meestal achterwege blijven. Talloze projekten zijn blijven steken in een fase, waarin ze naar de letter zijn opgeleverd, terwijl in werkelijkheid de gebruiker ontevreden is over de toepassing en de automatiseerders vinden dat dat voor een groot deel hun eigen schuld is.
Kortom, wanneer het projekt begint kan niemand bewijzen dat er iets fout zal gaan, achteraf blijkt er in negen van de tien gevallen heel veel te zijn fout gegaan.
Het probleem van de extra tijd tijdens het logisch ontwerp zou voor automatiseerders niet mogen bestaan. Gebruikers zouden moeten eisen dat er gewerkt wordt volgens bepaalde methoden. Automatiseerders en gebruikers zijn beide gebaat bij transaktie-ontwerp, dus zou de benodigde tijd geen probleem mogen zijn.
Bij dialoogsimulatie gaat het om drie trajekten waarvan de benodigde tijd moeilijk is te schatten:
- het maken van transaktieschema's. Er van uitgaand dat het niet alleen gaat om zeer lange komplexe transakties, kunnen er meestal vijf tot tien transaktieschema's per dag gemaakt worden. Die dag bestaat uit twee mandagen, een informatie-analistdag en een gebruikersdag.
- het voorbereiden van de simulatie. Gemiddeld zullen vijf tot tien schermlay-outs per dag kunnen worden voorbereid op de simulator door de informatie-analist, uitgaande van twee- tot vierhonderd tekens per scherm. Als het gemiddelde aantal schermen per transaktie geschat wordt, kan berekend worden hoeveel transakties per dag ongeveer kunnen worden voorbereid.
- het simuleren van de transakties door de gebruikers. Wanneer het om enkele gebruikers gaat zal men iedere transaktie circa vijf keer uitvoeren. Per dag kan men zo gemakkelijk vijf tot tien transakties simuleren. Het iteratieve aspekt is moeilijk te schatten. Misschien moet de informatie-analist een groot aantal schermen aanpassen. In de praktijk gebeurt dat niet vaker dan een tot twee keer. Bij grotere groepen gebruikers zal een kleine groep gebruikers fungeren als mede-ontwerpers, voor de overigen krijgt de simulatie meer het karakter van een demonstratie. Ook dan kan er natuurlijk nog kommentaar komen, maar meestal gaat het dan om details. Per dag kunnen, afhankelijk van het aantal simulatoren, vele transakties worden gesimuleerd.
Transaktie analyse verloopt in drie fasen:
- het maken van het detailschema aan de hand van transaktiesche-

ma's. Ook hier is de lengte van de transaktie weer het belangrijkste element bij de bepaling van de benodigde tijd. Als het transaktieschema gemiddeld twee pagina's lang is, kunnen er door een ervaren transaktie-analist meestal tien tot vijftien detailschema's per dag gemaakt worden.
- het invoeren van de detailschema's. Ervaren data entry-typistes kunnen dertig tot veertig detailschema's van de genoemde gemiddelde lengte intoetsen.
- het verwerken van de resultaten. De ergonomische konklusies worden meestal verwerkt in een rapport met enkele overzichten. In een paar dagen kunnen de konklusies ten aanzien van tientallen transakties op die manier worden vastgelegd. Technische resultaten, bijvoorbeeld ten aanzien van netwerkontwerp, vormen de input voor het ontwerpproces. Van de tijd die nodig is om een netwerk te ontwerpen vormt Transaktie analyse meestal maar een fraktie.
Het blijft een hachelijke zaak om cijfers te noemen voor de benodigde tijd, want eigenlijk bestaan er geen gemiddelde toepassingen. Bij twee bedrijven die orders via beeldscherm wilden invoeren werden in het ene bedrijf meer dan duizend orders per dag verwerkt, in het andere bedrjf was een medewerker een morgen bezig om een order in te voeren. In beide bedrijven heeft men het over het projekt ORDERENTRY.
De genoemde cijfers moeten als richtgetallen worden gezien. Planningen behoren niet door leken te worden gemaakt. Informatie-analisten en transaktie-analisten moeten transaktie-ontwerp een of twee keer hebben uitgevoerd om goede schattingen te kunnen maken in de gegeven situatie. De eerste paar keer moet er dus met wat extra tijd worden gekalkuleerd, op kosten van het inleerproces.

34.9 Responsetijden en transaktie-ontwerp

Na twintig tot dertig jaar automatiseringservaring in Europa worden nog dagelijks systemen opgeleverd met niet-akseptabele responsetijden. Nu is het probleem bij de responsetijden niet even simpel op te lossen, maar laten we eens enkele merkwaardigheden rond dit onderwerp op een rijtje zetten.
- De klachten over te lange responsetijden kan de gebruiker heel gemakkelijk bewijzen. Het meest a-technische direktielid heeft maar een korte demonstratie nodig, om te begrijpen waar het om gaat.
- Ook al zou 90% van alle interakties een prima responsetijd hebben, de gebruiker blijft klagen over die enkele interaktie, die achteraf inderdaad kritisch blijkt te zijn. Dit bewijst nog eens hoe zinloos het is om in specifikaties iets te schrijven over gemiddelde responsetijden.
- Er wordt achteraf onevenredig veel geld uitgegeven aan het ver-

beteren van responsetijden. Het begint meestal met meer geheugen, meer schijven en meer adviezen, wat vervelend is en duur maar relatief eenvoudig. In veel gevallen moet echter het database-ontwerp opnieuw gebeuren. Soms deugt het hele ontwerp niet en is er niets meer te verbeteren: terug naar af.

- Tijdens het logisch ontwerp gunt niemand zich de tijd om vast te stellen welke interakties kritisch zijn en wat dan de eisen zijn voor de responsetijd.

- Tijdens het technische ontwerp en de bouw is iedereen aan het werk alsof er nooit problemen met responsetijden zullen ontstaan. Dat is toch de enige mogelijke konklusie als blijkt, dat op een minicomputer de verwerking van een kritische interaktie 150 databasecalls omvat, zodat de responsetijd 45 seconden bedraagt?

Kortom, voor de meest irritante, de meest voorkomende en de duurste fout in de automatisering staan de eisen niet eens op papier. Wanneer het transaktie-ontwerp steeds korrekt is uitgevoerd, is per interaktie bekend wat de responsetijd maximaal mag zijn en welke verwerking daar bij hoort. Immers, op het transaktieschema staat de verwerking beschreven in gebruikerstermen.

De responsetijd bestaat maximaal uit twee delen: het transportdeel en het verwerkingsdeel. Het transportdeel is de tijd die nodig is om berichten te transporteren van beeldscherm naar computer en terug. Dat aspekt behoort tot het werkterrein van de transaktie-analist. Het verwerkingsdeel bestaat uit twee andere delen: de overhead van het operating system en de uitvoering van het applikatieprogramma. De overhead van het operating system moet beheerd worden door de systeemspecialist . Hij moet op de hoogte zijn van de overhead ten aanzien van het starten van programma`s en het uitvoeren van I/O operaties, de invloed van meer geheugen en meer schijven. Afgezien van overbelaste systemen is die overhead echter meestal niet het knelpunt. Dat zou betekenen dat computerleveranciers onbruikbare systemen leveren en dat is niet het geval. Wel verkopen ze soms, daartoe verleid door de konkurrentie, te kleine systemen, maar dat wordt hun ook vaak te gemakkelijk gemaakt! In het algemeen is het juist het ontwerp van de applikaties dat niet is afgestemd op het halen van bepaalde responsetijden. In ingewikkelde toepassingen kan het erg logisch klinken allerlei stuur- en beheerdatabases te bouwen, maar iemand moet die ontwerpen bekijken tegen de achtergrond van de responsetijden. Het is duidelijk dat dit het werkterrein is van systeemontwerpers, database-ontwerpers, programma-ontwerpers. Niemand kan van tevoren responsetijden berekenen, maar de wereld zou er al heel anders uitzien, als tijdens het technisch ontwerp werd vastgesteld dat aan bepaalde eisen lang niet voldaan kan worden. Het op een na laatste excuus, waar ontwerpers nog vaak mee aankomen is dat de gebruikers die funktie nu eenmaal zo gerealiseerd

wilden zien en dat er dus niets aan de logica van de applikatie viel te veranderen. Dat is prima als de responsetijden goed zijn. Wanneer echter de gebruiker een bepaalde funktie voorstelt of samen met de informatie-analist ontwerpt en meteen hoort dat de responsetijd 45 seconden zal bedragen, bedenkt hij zich nog minstens een keer. Het is overigens ook nog uitstekend wanneer de opmerkingen over lange responsetijden tijdens het technisch ontwerp worden gemaakt en teruggekoppeld worden naar de gebruiker. Dan is er nog steeds gelegenheid om naar alternatieven te zoeken. Het laatste veelgehoorde argument van ontwerpers is, dat het systeem goed was ontworpen, maar dat de gebruikers bleven komen met nieuwe eisen en nieuwe funkties. Daardoor moest het hele dataontwerp zovaak aangepast en uitgebreid worden, dat het nooit meer een goede performance kon leveren. Kennelijk zijn op het technische ontwerp nog veel wijzigingen geaksepteerd en aangebracht. Het zou trouwens niet voor het eerst zijn, als dit tijdens de bouw nog gebeurde. Het is duidelijk dat het hier niet gaat om een technisch probleem. Wanneer het logisch ontwerp, uitgevoerd met dialoogsimulatie, is afgerond, is inbreng van gebruikers afgelopen. Dat zullen ook gebruikers gemakkelijk aksepteren, wanneer ze met voldoende kollega's gewerkt hebben met de simulator. Maar zelfs wanneer een gebruiker te laat met een goed idee komt, tijdens het technisch ontwerp, dan zou het nog verwerkt kunnen worden. De prijs voor het te laat zijn is, dat een deel van het logisch en technisch ontwerp opnieuw gemaakt wordt. De tijd die dat kost is een fraktie van de tijd die nodig is om te lange responsetijden achteraf te verbeteren. Dat zijn geen technische problemen, maar zaken van goed projektmanagement, zowel aan de kant van de automatiseerders als aan de kant van de gebruikers. Hoe belangrijk goede kommunikatie tussen gebruiker en automatiseerders ook is, ze dient gekanaliseerd en beheerd te worden.
De aanpak ten aanzien van het voorkomen van problemen met responsetijden komt dus neer op:
- Tijdens het logisch ontwerp met behulp van dialoogsimulatie vaststellen welke interakties kritisch zijn voor de gebruiker en wat daarvan de maximale responsetijd is. Gebruiker en informatie-analist worden het daarover eens.
- Tijdens het logisch ontwerp de logische databases of gegevensmodellen optimaliseren op de toegangspaden naar de gegevens, die nodig zijn tijdens kritische interakties. Per interaktie dient een kompleet funktiemodel uitgewerkt te zijn. Wanneer dan al blijkt dat met het beste ontwerp de gewenste responsetijd nooit gehaald zal worden, vindt een voorgesprek plaats met de ontwerpers van de fysieke database. Blijkt uit dat gesprek dat ook de fysieke opslag geen oplossong kan bieden, vindt terugkoppeling naar de gebruiker plaats.

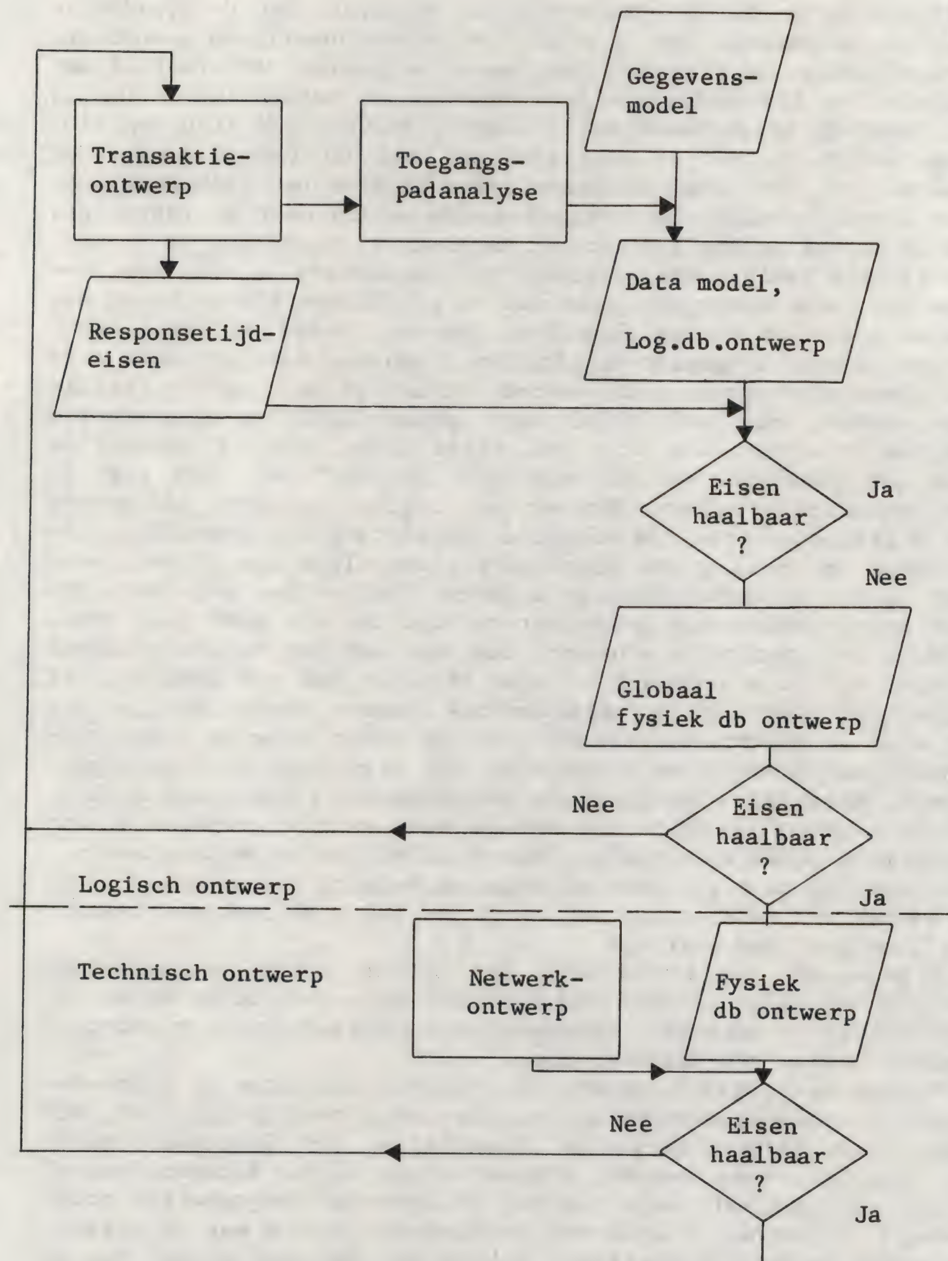


Fig. 34.9 Beheer van responsetijden.

- Tijdens het technisch ontwerp bepalen of de responsetijden gehaald zullen worden. Nogmaals, het gaat niet om marges van 10 of 20%. Er worden dagelijks technische ontwerpen gebouwd, waarvan de responsetijden een faktor tien tot vijftig hoger liggen dan was verwacht. Over de responsetijden valt de beslissing tijdens het technisch ontwerp. Daar moeten specialisten bewijzen dat de gestelde eisen gehaald kunnen worden. Daarvoor is een management nodig dat voldoende van het vak afweet om de informatie van de besproken vijf soorten vakmensen te beoordelen. Daar ligt het GO/ NO GO punt voor de bouw, wat de responsetijden betreft. Voor die beslissing moet de tijd genomen worden. De tijd die later nodig is om iets aan te lange responsetijden te doen, zal er een veelvoud van zijn. In Fig. 34.9 is een en ander nog eens in beeld gebracht, in Fig 34.10 zijn ook het programma-ontwerp en de omgeving erbij betrokken.

Samengevat, er zijn vier gebieden waar de oorzaak van te lange responsetijden kan liggen:

- het ontwerp van de gegevensopslag: Fig. 34.9
- het programma-ontwerp: zoveel mogelijk I/O's tijdens niet-kritische interakties.
- het netwerk: Transaktie analyse
- het systeem:
  - tuning: marginale verbeteringen
  - systeembelasting: onder andere Transaktie analyse

Tenslotte nog iets over responsetijden en de analysefase van een projekt. Toegegeven, het klinkt onwerkelijk om in een fase waarin alleen de bestaande situatie in kaart wordt gebracht, aan responsetijden te willen denken. Dan is er nota bene nog niet eens gekozen tussen batch- of interaktieve verwerking!

Laten we eens een bekende interaktieve toepassing bekijken. In de bestaande situatie zoeken gebruikers gegevens over een bepaald onderwerp op in overzichten. Laten we aannemen dat er per onderwerp een regel met informatie bestaat. Het gaat om een paar duizend onderwerpen. Er is dus een dik pak papier beschikbaar, waarin men dagelijks zit te bladeren. Kortom, een situatie die schreeuwt om automatisering. De ontwerpers hebben een prachtig systeem bedacht: de gebruikers typen de zoekargumenten in, het systeem zoekt de database af, maakt een bestand aan met alle relevante regels, sorteert het bestand en via het beeldscherm kan de gebruiker bladeren in het bestand en de regels zoeken waarin hij geinteresseerd is. Dit soort toepassingen wordt nog dagelijks gebouwd. Er is geen enkele ontwerper die verwacht dat hier de responsetijden flitsend zullen zijn. Verder is het typisch een oplossing die stamt uit de batch-wereld van twintig jaar geleden. We zullen maar buiten beschouwing laten hoe deze toepassingen de responsetijden op andere beeldschermen beinvloeden.

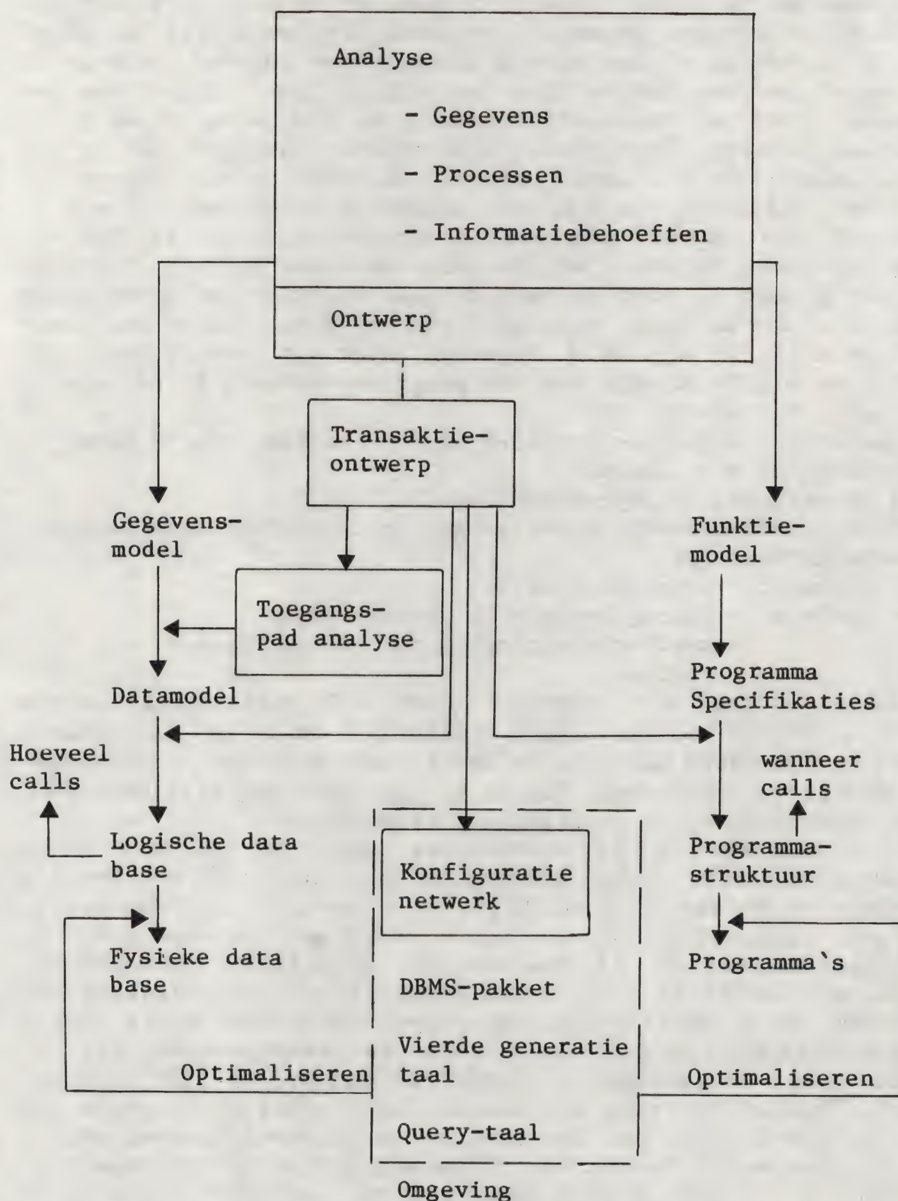


Fig. 34.10 Optimaliseren.

Toch is er op het ontwerp op zich niet veel aan te merken. Het had alleen nooit gebouwd mogen worden wanneer de gebruikers een responsetijd verwachten van enkele seconden. Dit soort toepassingen levert per definitie lange responsetijden op.
Het zou kunnen zijn dat er tijdens de analysefase al iets fout is gegaan. Te vlot is de bestaande situatie: bladeren in een stapel papier, vertaald naar een ontwerp: bladeren op een scherm. Wanneer in de analysefase nauwkeuriger was onderzocht op basis van welke kriteria de gebruiker zijn gegevens opzoekt, had de hele zoekoperatie van het systeem wel eens veel eenvoudiger kunnen zijn. Wanneer de kriteria nauwkeuriger worden aangegeven, wordt het aan te maken bestand kleiner, duurt het sorteren korter en hoeft er misschien helemaal niet meer gebladerd te worden op het beeldscherm. Kortom, er zijn veel situaties, waarin tijdens de analysefase al kan worden vastgesteld dat de bestaande handmatige situatie nooit tot een goed ontwerp kan leiden. Hoewel de analysefase nooit tevens ontwerpfase moet worden, mag er in het kader van de nauwgezetheid tijdens de analysefase best eens aan responsetijden gedacht worden. Soms zijn te lange responsetijden geen zaak van verkeerd ontwerp, maar van onzorgvuldige analyse.

Hoofdstuk 35
Dialoogsimulatie

35.1 Waarom dialoogsimulatie?

Dialoogsimlatie is een onderdeel van transaktie-ontwerp. Een transaktie is een procedure rond het beeldscherm, waarvan de dialoog een deel vormt. Meestal is dat het enige deel van de transaktie waar automatiseerders belang in stellen. De gebruiker krijgt te maken met het geheel van menselijke handelingen en de dialoog. Daarom vormt het gezamelijk, volgens een overeengekomen methode ontwerpen van transakties, de basis voor een goed verloop van een projekt. Het probleem van gebruikers die zelfs tijdens de bouw nog met nieuwe voorstellen komen is daarmee voor een belangrijk deel opgelost. Van hoog tot laag in de gebruikersorganisatie kan men het systeem beoordelen zoals het voor gebruikers zal werken. Dat systeem wordt besteld.
Dialoogsimulatie betekent voor de gebruikers: evaluatie van het eindprodukt tijdens het logisch ontwerp. Gebruikers zouden het dus zelf moeten willen, maar automatiseerders zijn eveneens gebaat bij deze vorm van samenwerking. In de delen voor informatie-analisten, transaktie-analisten en gebruikers wordt dat verder uitgewerkt.
Methoden voor kommunikatie met gebruikers maken bij automatiseerders een kans als ze aansluiten op de andere aspekten van de systeemontwikkeling. In het deel voor de informatie-analisten wordt

die aansluiting voor verschillende situaties uitgewerkt. Natuurlijk krijgen de ontwerpers hun schermlay-outs, maar nu als laatste fase van een evaluatieproces waarin transakties zijn ontworpen. Transaktie-analisten krijgen uitgekristalliseerde transaktieschema's voor Transaktie analyse.
Voor het uitvoeren van dialoogsimulatie als onderdeel van transaktie-ontwerp zijn nodig:
- gebruikers die zijn voorbereid op het funktioneren als medeontwerper, zie deel 6 en (15),
- informatie-analisten die verder kijken dan beeldschermlay-outs en deel 4 hebben gelezen,
- een dialoogsimulator, bijvoorbeeld op een personal computer (22),
- een planning, waarin tijd is gereserveerd voor transaktie-ontwerp door automatiseerders en gebruikers.

Tijdens de analyse is er al enig uitzicht ontstaan op transakties, zoals dat is beschreven in het deel voor de informatie-analisten in de paragraaf Van analyse naar transakties. (42.2)
Voorbereide gebruikers en informatie-analisten leggen de toekomstige procedure aan het beeldscherm vast op een transaktieschema. Daarmee is de relatie tussen de dialoog en andere handelingen van de gebruiker vastgelegd op een voorlopig transaktieschema. Waar iets op een scherm verschijnt is helemaal nog niet van belang. Er is nu een transaktie ontstaan met een naam, een werkplek en een duidelijk gezicht naar de gebruiker en naar de informatie-analist. Daarmee is dan ook de basis gelegd voor een betere kommunikatie ten aanzien van cijfers, frequenties en pieken via de resultaten van Transaktie analyse. De informatie-analist brengt nu de transaktie tot leven door op de dialoogsimulator de dialoog vast te leggen. Daarvoor hoeven slechts schermlay-outs te worden gemaakt en funkties van velden te worden aangegeven. Er hoeft niet te worden geprogrammeerd en er hoeven geen bestanden te worden aangemaakt.
De simulator is voor de gebruiker een beeldscherm en een toetsenbord en dat is voldoende om de toekomstige situatie volledig te simuleren kompleet met brondokumenten, balies of telefoongesprekken. Dat de simulator niet precies gelijk is aan het uiteindelijke beeldscherm is voor gebruikers geen enkel probleem. In de eerste plaats duurt het nog maanden voordat de toepassing gebouwd is en vergeet hij dus hoe het toetsenbord van de simulator er precies uitzag. In de tweede plaats liggen problemen van gebruikers niet op het niveau van de kleur, de vorm of de plaats van de toetsen. Dialoogsimulatie is geen terminalemulatie.
Dialoogsimulatie geeft gebruikers de gelegenheid de toekomstige situatie te evalueren en de automatiseerders een goede basis voor en aansluiting op andere ontwerpaktiviteiten.

35.2 Dialoogsimulatie versus prototyping

Een prototype is het eerste model. De produktielijn levert later honderden of duizenden produkten af die gelijk zijn aan het prototype. Een prototype wordt gebouwd, omdat het beter is nog wat wijzigingen aan te brengen in het prototype, dan in een gereed produkt en de produktielijn zelf. In de automatisering speelt het produktie-aspekt meestal niet, omdat we bijvoorbeeld waar een voorraadbeheersysteem nodig hebben. Wel is het zo, dat wijzigingen in het produkt vaak zeer kostbaar en tijdrovend zijn. Daarom probeert men een systeem te bouwen dat, zonder dat het eigenlijke systeem er is, toch model kan staan voor dat systeem. Dat prototype moet dus funktioneren als het echte systeem, maar ook weer niet te echt, want dan hebben we het uiteindelijke systeem gebouwd! Dat betekent dat er een computersysteem moet zijn, beeldschermen, programmatuur, bestanden, enzovoort.
In de praktijk komt het er op neer dat men een subset van het systeem bouwt. Bijvoorbeeld alleen de "mooi weer situatie". Er kunnen op het beeldscherm wel gegevens worden ingevoerd, maar niet alle kontroles op juistheid en kompleetheid worden doorgevoerd. Er zijn wel gegevens beschikbaar, maar niet alle bestanden zijn volledig gevuld.
Het doel van prototyping is de gebruiker een zo goed mogelijk inzicht te geven in wat hij straks kan verwachten. Hij kan daar nu nog kommentaar op leveren, andere eisen stellen, met nieuwe voorstellen komen. Dat werkt natuurlijk alleen als het prototype snel kan worden aangepast. Dat betekent dat de software geschreven is in een programmeertaal van hoog niveau, die snel te schrijven is en snelle aanpassingen mogelijk gemaakt. Vandaar dat in de literatuur in dit verband gesproken wordt van programmageneratoren. Anderzijds brengen leveranciers van generatoren hun produkten graag in verband met prototyping. `t Is een beetje een modewoord: bij bedrijven met grote automatiseringsafdelingen is het weer eens wat anders dan database design en structured programming. In het midden- en kleinbedrijf gunt men zich meestal geen tijd voor dit soort zaken. Wil prototyping zin hebben, dan moet gerekend worden met 10% van de projektkosten.
Bij prototyping van gegevensverwerkende systemen gaat het om drie aspekten: gegevens, funkties (algoritmen) en dialogen.
Men kan een prototype maken van alle gegevens en gegevensstrukturen. Wanneer men vervolgens de funkties die de gebruiker wenst, loslaat op die gegevens kan men het resultaat beoordelen. Hoe die resultaten gepresenteerd worden is niet van wezenlijk belang. Ze zouden op lijsten afgedrukt kunnen worden.
Tenslotte kan men de dialoog tussen gebruiker en systeem bouwen

en aan de gebruiker tonen. Het geheel van de delen geeft natuurlijk een getrouw beeld van het te bouwen systeem. Kortom, van de toekomstige situatie wordt een zo getrouw mogelijk beeld opgebouwd. Hoe ver men daarmee wil gaan blijft altijd ter diskussie. Vast staat in ieder geval, dat er minstens een konfiguratie nodig is waarop het prototype wordt gebouwd. Wanneer het om nieuwbouw gaat, zal het duidelijk zijn dat computerleveranciers zich inspannen om aan te tonen dat zij beschikken over alle middelen voor prototyping. Immers, wanneer dat als verkoopargument effekt heeft, is er weer een systeem geplaatst. De kans dat de koper na de prototyping fase het werkelijke systeem op een computer van een andere leverancier gaat bouwen, is natuurlijk niet erg groot. Laten we eerst de technische kant vergelijken en vervolgens de gebruikersaspekten.

Bij prototyping moeten beschikbaar zijn: gegevens, programma`s, de dialoog via een beeldscherm en een konfiguratie. Van de gegevens zullen wel alle types, maar niet alle occurrences nodig zijn. Als het systeem werkt voor 10 records zal het ook wel werken voor 1000, indien we ons tenminste uitsluitend bezighouden met de logica van de toepassingen. Daarnaast ontstaat al heel gauw een diskussie over de vraag welke bestanden beslist nodig zijn.

Vervolgens zijn er programma`s nodig die de bewerkingen uitvoeren, zoals die op de een of andere wijze zijn gespecificeerd. De beslissing over wat er nu wel en wat niet geimplementeerd moet worden, is nog veel meer aanvechtbaar dan die over de gegevens. Men ziet bijvoorbeeld meestal af van allerlei kontroles op de input en de bijbehorende foutsituaties. Dat is nog eenvoudig bij numerieke en niet numerieke input. Wanneer het gaat over foutsituaties die de afloop van de dialoog of het hele verloop van de transaktie beinvloeden, wordt het al moeilijker vast te stellen welke aspekten wel en welke niet geimplementeerd moeten worden. Hoe men het ook bekijkt: er moeten programma`s gebouwd worden. Hoewel er nog steeds programmeurs zijn, die geloven in snelle houtje-touwtjeprogramma`s, is de gemiddelde automatiseringsafdeling wel achter, dat dit onder andere problemen oplevert met betrekking tot de leesbaarheid en de onderhoudbaarheid van de software. De essentie van prototyping-software is uiteraard de flexibiliteit ten aanzien van wijzigingsvoorstellen van gebruikers. Met andere woorden: de onderhoudbaarheid moet veel beter zijn dan die van operationele software. Vandaar, dat men werkt met programmeertalen van hoog niveau of met software-generatoren. Maar hoe men het ook doet, er worden hoge eisen gesteld aan de manier van programmeren. Wanneer men het eens is geworden over gegevens en programmafunkties, kan de dialoog exact gebouwd worden volgens de specifikaties. Als er voldoende "mappingtools" beschikbaar

zijn, kunnen schermlay-outs worden gemaakt die gekoppeld worden aan de programma's. En daarmee is het automatiseringsplaatje kompleet.
Bij dialoogsimulatie gaat het alleen om de dialoog. Aan het einde van het simulatieproces is niets bekend over programmafunkties en de daarbij behorende gegevens. Wanneer echter dialoogsimulatie een onderdeel is van het transaktie-ontwerp, ligt de situatie heel anders. Dan zijn er eerst transaktieschema's gemaakt, waarop wordt aangegeven om welke funkties en gegevens het gaat bij de verwerking door de computer.
Een ander belangrijk aspekt van dialoogsimulatie is dat er niet geprogrammeerd hoeft te worden. In de fase "logisch ontwerp" heeft de informatie-analist geen andere deskundigen nodig. Hij is alleen met de gebruiker en de simulator. Hij ontwerpt transakties, maakt de schermlay-out en definieert de dialoog. Bij gebruik van een goede dialoogsimulator is hij in een dag ingewerkt. Er is geen sprake van een projektaanpak, programmeringsdeskundigheid, bestandsbeheer of software-releases. Zonder veel moeite kunnen ter plekke verscheidene alternatieven voor een dialoog worden gebouwd en "geprototyped". Bij dialoogsimulatie zullen altijd kleine afwijkingen van de werkelijkheid blijven bestaan. Het heeft alles met het vakmanschap van de informatie-analist te maken, hoe hij daarmee omgaat. Hij moet namelijk beoordelen in hoeverre die verschillen werkelijk iets te maken hebben met problemen van gebruikers.
Daarmee zijn we terechtgekomen bij de gebruikersaspekten van dialoogsimulatie. Meestal is de simulator niet identiek met het toekomstige beeldscherm. In de praktijk blijkt het te gaan om verschillen die een gebruiker gemakkelijk aksepteert of waarvan hij zich gemakkelijk een beeld kan vormen. Het hoofddoel van dialoogsimulatie als onderdeel van het transaktie-ontwerp blijft het voorkomen van problemen achteraf. Het gaat daarbij om problemen op het gebied van de ergonomie. Daarbij valt te denken aan zaken als:
- de gebruiker begrijpt de terminologie op het beeldscherm niet,
- hij kan de betekenis van allerlei kodes niet onthouden,
- hij weet in bepaalde situaties niet hoe hij verder moet,
- hij had ander eisen gesteld als hij beter had begrepen hoe een interaktieve toepassing funktioneert.

Dat is wat anders dan bijvoorbeeld de plaats van de funktietoetsen op het toetsenbord. Misschien is het toetsenbord van de dialoogsimulator anders ingedeeld dan dat van het uiteindelijke beeldscherm, maar dat mag nog geen reden zijn om dialoogsimulatie te verwerpen. Over die zaken gaat het niet tijdens de evaluatie van interaktieve toepassingen.
Voor de gebruiker achter een beeldscherm is het verschil tussen

	Prototyping	Dialoogsimulatie
	Automatiseringsaspekten	
Data :	Alle gegevens, enkele occurrences.	Standaard attributen, enkele occurrences.
Programma's:	Hoofdlijnen, weinig opvang van fouten.	Standaard funkties.
Dialoog :	Beeldschermlay-out.	Transaktie-ontwerp. Beeldschermlay-out. Dialoogdefinitie.
Hardware :	Komplete konfiguratie.	Portable micro.
	De informatie-analisten moeten beschikken over specialistische kennis, en anders is ondersteuning van anderen noodzakelijk.	Iedere informatie-analist leert in een dag de simulator bedienen.
	Gebruikersaspekten	
	Dialoogontwerp.	Transaktieschema. Dialoogontwerp.
	Werken met beeldschermen Voorstellen voor (snelle) wijzigingen. Realisering volgens software releases.	Werken met beeldschermen. Dialoog direkt te wijzigen.
	Dummy-aktiviteit.	Dummy-aktiviteit.
	Pas te realiseren wanneer hardware, software en know-how beschikbaar zijn.	Op elk moment te realiseren.

Fig. 35.1 Prototyping versus dialoogsimulatie.

prototyping en dialoogsimulatie erg klein. Hij "ziet" geen programma`s, geen bestanden of databases. Bij dialoogsimulatie doen de gegenereerde programma`s weinig meer dan attributen lezen en displayen op het scherm. De ingetypte informatie wordt bijna nooit verwerkt. Stel dat de gebruiker twee bedragen moet intypen en de computer de som moet displayen. Na het intypen van de bedragen en het verlopen van de responsetijd displayed de simulator een willekeurig bedrag. Bij sommige automatiseerders lopen dan de rillingen over de rug, maar de eenvoudigste gebruikers hebben er geen problemen mee. We hebben het uiteraard over gebruikers die via een presentatie over dialoogsimulatie goed zijn voorbereid. Het feit dat de simulator niet eens twee bedragen bij elkaar op kan tellen, werkt positief. De gebruiker begrijpt eens te meer dat de uiteindelijke programma`s nog ontwikkeld moeten worden. De gegevens die gedisplayed worden, komen uit de standaard attributenpot van de simulator, bij prototyping worden de bestanden gevuld met bedrijfsgegevens. Bij dialoogsimulatie is een standaard attributenpot beschikbaar, zodat er altijd meteen gesimuleerd kan worden. Hoe kort de voorbereidingstijd ook is, als de dialoog is gedefinieerd, werkt de simulator. Bij een goede simulator is het mogelijk attributen aan te passen aan het projekt of het bedrijf. Maar ook hier geldt dat gebruikers meestal veel minder problemen hebben met onaangepaste attributen dan automatiseerders.
Zowel voor prototyping als voor dialoogsimulatie geldt, dat het de gebruiker om "dummy"-aktiviteiten gaat als hij achter het beeldscherm zit. Het werk heeft nog geen zin, het gaat om de beleving van interaktieve toepassingen.
Bij dialoogsimulatie gaat het altijd alleen om de dialoog. Wat op de printer of op de plotter verschijnt, valt buiten de simulatie. Bij prototyping is in principe alles te programmeren en als het prototypingsysteem hetzelfde systeem is als het uiteindelijke systeem, dan kunnen alle aspekten geprototyped worden. Dialoogsimulatie is echter niet opgezet om alles zo echt mogelijk te laten funktioneren, maar om de grote massa toepassingen die vanwege de interaktie tussen mens en computer problemen opleveren, tijdens het ontwerp dichter bij de gebruiker te brengen.
We kunnen bovenstaande beschouwing in kaart brengen als in Fig. 35.1 is aangegeven.

Karakteristieken van prototyping en dialoogsimulatie.

In (4) wordt een aantal karakteristieken van prototyping genoemd. We zullen ze nog eens op een rijtje zetten en dan prototyping vergelijken met dialoogsimulatie.
- Lage kosten.
Bij prototyping worden de kosten door een groot aantal faktoren

bepaald en hangen ze af van de situatie. Wanneer het systeem waarop prototyping moet worden uitgevoerd nog moet worden aangeschaft, wordt het kostenplaatje erg ondoorzichtig. Wordt prototyping uitgevoerd op een bestaand systeem waarop ook de uiteindelijke applikaties gaan draaien, dan kan het kostenplaatje beperkt blijven tot de projektkosten. Die kosten worden op dezelfde manier berekend als van elk ander automatiseringsprojekt, want het gaat hier om precies dezelfde aspekten: analyse, ontwerp, bouw, test, invoeren en evaluatie. In de praktijk wordt 10% van de totale projektkosten als vuistregel gehanteerd.
Bij dialoogsimulatie worden de kosten bepaald door de aanschaf van een microcomputer, als die al niet aanwezig is, en het simulatiepakket. Het analysetrajekt zal net zo verlopen als bij prototyping. In de ontwerpfase gaat het uiteraard alleen om de dialoog. In een dag kunnen diverse transakties op de simulator ontwikkeld worden. De trajekten ontwerp, bouw, test, invoeren en evaluatie worden uitgevoerd in enkele uren. Van invloed is natuurlijk het aantal gebruikers dat men erbij wil betrekken, maar dat geldt ook bij prototyping. Wanneer men bij de kostenbepaling ook nog de opleiding van informatie-analisten of prototypingprogrammeurs meeneemt wordt het verschil tussen beide methoden nog groter. Uiteraard is het onvoldoende alleen naar de kosten te kijken, omdat volledige prototyping veel meer oplevert dan dialoogsimulatie.

- Flexibiliteit.

Als het goed is, is de software in een programmeertaal van hoog niveau geschreven. Eigenlijk zou het niveau zo hoog moeten zijn dat een informatie-analist zijn ontwerp rechtstreeks kan invoeren. Dan zou wijzigen ook niet moeilijk zijn. Een belangrijk aspekt van prototyping is immers, dat we het iteratieve aspekt van het ontwerpen samen met gebruikers inhoud willen geven. Bouw iets, laat hen er mee werken, evalueer, pas het ontwerp aan, laat hen er weer mee werken. Dezelfde flexibiliteit moet ook aanwezig zijn bij het gegevensontwerp en het dialoogontwerp. Hoe nauwkeuriger men wil prototypen hoe meer programma's en bestanden er ontstaan. Al heel snel ontstaat hetzelfde probleem als bij elk automatiseringsprojekt: hoe houd ik de wijzigingen in de hand.
Dat betekent in de praktijk: releases van de prototyping-software.
Bij dialoogsimulatie ziet de gebruiker hetzelfde als bij prototyping: een beeldscherm waarop de dialoog "live" werkt. Aangezien de informatie-analist noch gegevensmodellen noch programma's heeft ontworpen kan elke wijziging ter plekke worden uitgevoerd en worden beoordeeld. Voor de gebruiker is de flexibiliteit veel groter dan bij prototyping.

- Participatie.

Vanzelfsprekend is zowel bij prototyping als bij dialoogsimulatie

de betrokkenheid van de gebruiker groot. Zeker als we uitgaan van hetzelfde niveau van vakmanschap van de informatie-analist. Wat we daarmee bedoelen staat in de paragraaf "4 x M". Als dialoogsimulatie onderdeel is van transaktie-ontwerp , kan als vervolg op de simulatie de gebruikerssituatie ook nog kwantitatief in beeld gebracht worden. Dan ziet hij bijvoorbeeld hoeveel uur per dag hij achter het beeldscherm zit. Daardoor zal zijn betrokkenheid en die van personeelszaken zeker toenemen.

- Snelheid.

Het is een bekend feit dat de gemiddelde gebruiker geen idee heeft van het werk van automatiseerders. Ze vinden alleen dat het altijd zo lang duurt voor ze iets "tastbaars" krijgen om mee te werken. Wanneer de hulpmiddelen voor prototyping effektief zijn en men zich beperkt tot het prototypen van de belangrijkste aspekten, dan kan men de gebruiker heel snel iets laten zien. Doordat er programma's ontwikkeld zijn en echte bestanden zal het geheel een echte indruk maken. De doorlooptijd is kort in vergelijking met die van een echt projekt. Het probleem daarbij is wel, dat de gebruiker gemakkelijk een verkeerd beeld krijgt van de doorlooptijd van zijn projekt.
Bij dialoogsimulatie is de snelheid veel groter omdat in enkele dagen heel wat transakties gebouwd kunnen worden. Evenals bij prototyping, gaan we bij dialoogsimulatie uit van gebruikers die goed voorbereid zijn. Dan zal de gebruiker begrepen hebben dat er helemaal geen programma's en bestanden gebouwd zijn. Mocht hij dat weer vergeten zijn, dan wordt hij daar tijdens de simulatie steeds weer aan herinnerd. Als de simulator de som van twee ingetypte bedragen moet displayen, verschijnt er een willekeurig bedrag.

- Eenvoudig te leren.

Dat is inderdaad een eis die gesteld moet worden aan de gebruikte programmeertaal. Maar een informatieanalist is nu eenmaal geen programmeur. Zeker als het om wat ingewikkelde applikaties gaat of om een grote hoeveelheid programma's, zal hij snel afhaken. En dat betekent dat er toch een senior-programmeur aan te pas moet komen. En daarmee zijn we weer in bekend vaarwater: de kommunikatie tussen de informatie-analist, de systeemontwerper en de programmeur. In projekttermen uitgedrukt: analyse, logisch ontwerp, technisch ontwerp en bouw. Wanneer de prototypingtaal niets gemeen heeft met de taal van de uiteindelijke applikaties, dan is het leren van de prototypingtaal dus gewoon een extra inspanning die men zich moet getroosten.
Bij een goede dialoogsimulator is een informatie-analist in een dag ingewerkt. Hij doet niets anders dan de schermlay-out maken en de dialoog vastleggen.

- Beperkte opzet.

Men zal een keuze moeten maken tussen wat wel en wat niet in het prototype wordt opgenomen. Meestal kiest men voor de meest voorkomende transakties en dus voor de mooi weer situatie. Er wordt een subset gedefinieerd van het gegevensmodel.
Bij dialoogsimulatie kost het geheel maar zo weinig tijd, dat men gemakkelijk veel meer transakties kan simuleren. Maar ook hier zal men zich zeker enige beperkingen opleggen. Zo zou men best alle foutsituaties kunnen simuleren. Wanneer de gebruiker echter in enkele transakties gezien heeft hoe er door het systeem gereageerd wordt op fouten, dan hoeft hij daarna niet alsnog alle foutboodschappen op het scherm te zien verschijnen. De teksten van die boodschappen kan hij ook op papier uitstekend beoordelen.
- Proefomgeving.
Enigzins afhankelijk van de configuratie zal er een proefopstelling moeten worden gemaakt. Daar zullen diverse gebruikersgroepen gekonfronteerd worden met de transakties op beeldschermen. Beter is het, de eigen werkplek te voorzien van een beeldscherm. Dat zal vaak niet gebeuren, omdat dan de bekabeling steeds moet worden gewijzigd. Bovendien zal bij gedecentraliseerde bedrijven niet altijd de mogelijkheid aanwezig zijn om op elke lokatie een verbinding met de computer tot stand te brengen. Maar, zoals gezegd, het hangt nauw samen met de beschikbare configuratie en het het budget. Bij dialoogsimulatie gaat het om een beeldscherm dat tegelijk de computer is. De microcomputer kan dus overal mee naar toegenomen worden. De informatie-analist reist met een microcomputer in zijn auto langs de diverse vestigingen. Dan kan iedereen in zijn eigen omgeving en desgewenst op zijn eigen werkplek simuleren. Als er op de vestigingen al microcomputers aanwezig zijn, vervoert de informatie-analist alleen nog een paar floppies! Bij dialoogsimulatie is men volledig onafhankelijk van de soort configuratie en de beschikbaarheid ervan.
- Deskundige begeleiding.
Bij prototyping gaan dan al gauw de gedachten uit naar de supersnelle hoofdprogrammeur. De gebruiker hoeft maar een half woord te zeggen en hij heeft de programma's al gewijzigd. Of die programmeur ook de goede informatie-analist is, is natuurlijk de vraag. Hier is eigenlijk behoefte aan gemengd vakmanschap: informatie-analist en programmeur. Uit oogpunt van personeelsbeleid meestal niet zo gemakkelijk in een persoon te verenigen.
Bij dialoogsimulatie wordt niet geprogrammeerd en de informatie-analist verzorgt de simulatie van het begin tot het eind. Hij kan zich koncentreren op het wezenlijke van zijn vak. Hij hoeft geen programmeertaal te leren en hij hoeft niet te kommuniceren met een programmeur.

Plussen en minnen.

Bij prototyping is een programmeertaal van hoog niveau gewenst, zo niet noodzakelijk. Wanneer aan die voorwaarde is voldaan zijn er geen beperkingen in de simulatie voor de gebruiker. Het geheel kan draaien op goedkope snelwerkende apparatuur: een bestaand systeem, een nieuw systeem of een micro. Wanneer speciale hardware wordt aangeschaft moet algemeen bruikbaar zijn. Prototyping heeft als methode geen enkele relatie met het netwerkontwerp.
Problemen die kunnen ontstaan:

- Gevoel van vrijblijvendheid bij de gebruikers. Het kan immers altijd nog gemakkelijk veranderd worden.
- De gebruiker kan een verkeerd idee krijgen van de doorlooptijd van een projekt. Er wordt per slot van rekening een werkend systeem gebouwd.
- Er wordt teveel vertrouwd op de gebruiker.
  Een gebruiker bouwt nooit een optimaal systeem. Zoals in dit boek vaker is betoogd gaat het juist om de kombinatie van het vakmanschap van de gebruiker en dat van de automatiseerder.
- Wat is deskundige begeleiding? In een prototyping-omgeving is dat eerder een ervaren programmeur dan een informatie-analist.
- Hoe vaak wordt het systeem nu eigenlijk gebouwd?
- Wanneer komt er een einde aan de inspraakperiode? Anders gezegd: waneer wordt de prototypingfase afgesloten? Is er tijdens de rest van het "normale" logische ontwerp geen inspraak meer mogelijk?

Bij dialoogsimulatie liggen de aksenten anders. Er wordt niet geprogrammeerd, er is dus geen krachtige programmeertaal nodig. Dat betekent echter ook dat er geen programmafunkties worden nagebootst. Die programmafunkties zijn alleen aangeduid op het transaktieschema. Als de simulator een algemeen inzetbare micro is, bestaat er een verschil met de uiteindelijke terminal. Dialoogsimulatie is geen terminalemulatie. Wanneer dialoogsimulatie deel uitmaakt van transaktie-ontwerp dan is er, via transaktie analyse, een duidelijke koppeling met het netwerkontwerp. Bij dialoogsimulatie wordt de gebruiker er steeds aan herinnerd dat de simulator nog lang geen computer is. Hij kan nog niet eens twee ingetypte bedragen bij elkaar optellen. De gebruiker krijgt dus niet zo gauw een verkeerd idee van de doorlooptijd van een projekt: de dialoog is maar een klein deel van een projekt. De deskundige begeleiding is volledig in handen van een informatie-analist die zowel de methode als de bediening van het gereedschap beheerst. Wanneer dialoogsimulatie is afgesloten is voor de gebruiker het ontwerpen van het interaktieve deel van de toepassing afgesloten. Alle te laat ingebrachte ideeen worden meegenomen in een tweede release van de te ontwikkelen software. Bij dialoogsimulatie zul-

len er in de praktijk een aantal beperkingen optreden die bepaald worden door het verschil tussen de hardware van de simulator en de hardware van het uiteindelijke systeem. Zo kan het dat de beeldschermen die later worden toegepast een scroll-funktie hebben en dat de simulator dat iet kan nabootsen. Dit is een typisch voorbeeld van een onbelangrijk verschil. Wanneer aan een gebruiker een keer om welk computersysteem dan ook, wordt gedemonstreerd wordt wat scrolling inhoudt, hoeft dat niet meer gesimuleerd te worden. Toch blijkt in de praktijk steeds weer dat de automatiseerders moeilijk over de verschillen heen stappen. In feite zoeken ze naar terminalemulatie. Dat gaat met prototyping op het uiteindelijke systeem uiteraard het beste.
Wanneer dialoogsimulatie op zichzelfstaand wordt uitgevoerd, bestaat het resultaat uit schermlay-outs en de eisen voor de responsetijden. Aangezien er standaard attributen worden gedisplayed, is er geen relatie met de gegevens in modellen of bestanden. Bij een groot aantal transakties kan dat een bezwaar zijn. Er zijn twee mogelijkheden om dit probleem op te lossen. Dialoogsimulatie behoort deel uit te maken van transaktie-ontwerp. Dat betekent dat er voorafgaand aan de simulatie een transaktieschema wordt gemaakt. Op dat scherm worden de gegevensnamen vermeld. Alle transaktieschema's dienen vergeleken te worden met de aanwezige gegevensmodellen of data dictionary. Daarnaast zal een goede dialoogsimulator over de mogelijkheid beschikken om per veld, waarop een attribuut wordt gedisplayed, een gegevensnaam op te geven. Wanneer een lijst geprint wordt van die namen, kunnen alle gegevensnamen in de simulator vergeleken worden met gegevensmodellen of met de data dictionary.
Bij de keuze tussen dialoogsimulatie en prototyping dienen de voor- en nadelen in een bepaalde bedrijfssituatie nauwgezet tegen elkaar te worden afgewogen. Hoe verschillend men ook kan oordelen over de resultaten van beide methoden, over de effektiviteit ten aanzien van het dialoogontwerp, de korte doorlooptijd en de lage kosten van dialoogsimulatie, is iedereen die ermee gewerkt heeft, het eens.

Konklusies.

- Het verschil tussen prototyping en dialoogsimulatie is vaak erg klein voorzover het om het dialoogontwerp gaat.
- Dialoogsimulatie is onder alle omstandigheden sneller en gemakkelijker dan prototyping.
- Bij dialoogsimulatie zijn geen gegevens of programma's betrokken. Daarom kan dialoogsimulatie veel eerder in de projektaanpak worden uitgevoerd dan prototyping.
- Dialoogsimulatie kan dienen als voor-trajekt voor prototyping.

- Voor beide methoden bestaan er enige voetangels en klemmen:
  - voorbereiding van de gebruikers
  - selektie van deelnemende gebruikers
  - verwachtingen van de gebruikers
  - het vakmanschap van de informatie-analist blijft de bepalende faktor.
- Misschien zijn beide methoden niet goed vergelijkbaar. Voor een gebruiker die niets van bestanden en verwerking door een computer weet, is dialoogsimulatie praktisch gelijk aan prototyping. Voor de automatiseerders is het dialoogontwerp maar een fraktie van het hele gebeuren en dialoogsimulatie daarom onvergelijkbaar met prototyping.
- Gezien de in vorige paragraaf behandelde aspekten van beide methoden kan worden vastgesteld, dat dialoogsimulatie in elk geval en te allen tijde kan worden toegepast, onafhankelijk van de aanwezigheid en de beschikbaarheid van bestaande systemen.
- Dialoogsimulatie kan worden ingepast in elke bestaande systeemontwikkelingsmethode en projektaanpak. In tegenstelling tot de situatie bij prototyping kan de bestaande projektstruktuur, -organisatie en bemanning per fase ongewijzigd blijven.

Hoofdstuk 36
Transaktie analyse

36.1 Waarom Transaktie analyse?

Zonder in te gaan op de details van Transaktie analyse zullen we de argumenten bespreken die pleiten voor de toepassing van deze analysemethode. Die argumenten sluiten aan bij de resultaten. Transaktie analyse levert twee soorten resultaten op. De eerste soort heeft te maken met de voorkant van het beeldscherm, de tweede soort net de netwerkkant ervan.
Het gaat dan om het ontwerp van de nieuwe transaktie. Er zijn vaak enige alternatieven, zoals blijkt uit het volgende voorbeeld.
Stel dat er op een verkoopkantoor schriftelijk en telefonisch orders binnenkomen. Beide worden genoteerd op orderformulieren. De medewerkers behandelen alles, zowel schriftelijke als telefonische orders. Deze vastlegging van orders moet in de toekomst via een beeldscherm worden gerealiseerd. Er kunnen twee totaal verschillende transakties worden ontworpen. Immers de ingave van een order vanaf een dokument is geheel anders dan de invoering van een order die telefonisch binnenkomt. In de eerste plaats is het al een probleem welke klantidentifikatie wordt ingetypt. Bij schriftelijke orders is enige werkvoorbereiding in de vorm van het bepalen van klantnummers mogelijk. Bij telefonische orders moet met een direkt in te voeren zoekargument worden gewerkt. In

de tweede plaats is de volgorde van orderregels bij telefonische orders onvoorspelbaar. Bij schriftelijke orders kan de schermlay-out gelijk zijn aan het vroegere orderformulier. Een heel ander aspekt is het inwerken van de toekomstige gebruikers. Wie kwantificeert wat er in de aanloopfase fout kan gaan? Wie rekent voor welke achterstand er kan ontstaan in pieken? Wie rekent voor wat het in aantal beeldschermen uitmaakt of telefonische orders direkt worden ingevoerd of eerst worden vastgelegd op een orderformulier en dan massaal worden ingevoerd.
Transaktie analyse levert reeds tijdens het logisch ontwerp de cijfers op om een weloverwogen keuze te maken uit de alternatieven. Daarbij is het mogelijk om de gebruiker te laten kiezen uit de diverse mogelijkheden. Niet op grond van technisch, voor hem onbegrijpelijk cijferwerk maar op grond van getallen uit zijn denkwereld: aantal orders per dag, overwerk tijdens pieken, manier van werken. Dan worden de gesprekken over sociale aspekten van automatisering weer een stuk konkreter. Daarnaast kan de gebruiker worden voorgerekend wat de konsekwenties zijn van onnauwkeurige gegevens over aantallen transakties, pieken en dergelijke. Op die manier worden problemen van de gebruiker verwezen naar de plaats waar ze horen: bij de gebruiker.
Met de kengetallen van Transaktie analyse als basis kunnen wat het werken met het beeldscherm betreft de volgende facetten worden gekwantificeerd:

- Het benodigde aantal terminals.

Dat is het aantal terminals om de throughput van een afdeling te realiseren. Met ander woorden nu kan reeds in het logische ontwerp de grootte van het terminalpark worden vastgesteld. Soms wordt het aantal beeldschermen door geheel andere zaken bepaald. Bijvoorbeeld door het aantal bureau`s of het aantal loketten. Nu zal het voor de gemiddelde niet veel uitmaken of een bepaald aantal transakties nu met 10 of met 20 beeldschermen wordt uitgevoerd. De claim op de resources ligt immers vast per transaktie. Wat wel van groot belang kan zijn is het feit dat in pieken ook de "overtollige" terminals in eens bezet zijn. Dan krijgt de computer dus wel te maken met de piekbezetting van alle beeldschermen. Die invloeden worden nu kwantificeerbaar.

- De lengte van de wachtrij.

Bij lokettransakties is de lengte van de wachtrij van belang. Sommige bedrijven hebben eigenlijk maar een probleem met de automatisering: service aan de klant. De lengte van de wachtrij is daarom van essentieel belang. Transaktie analyse levert de cijfers om met behulp van de wachtrijtheorie deze lengte te bepalen.

- De benodigde tijd.

Hoeveel tijd kosten alle ontworpen transakties per werkplek? Tijdens het logisch ontwerp worden er een aantal transakties ontwor-

pen voor een bepaalde werkplek. Soms voor een groot deel nieuwe transakties omdat de automatisering wel eens nieuwe mogelijkheden biedt. De vraag die dan naar voren komt is: hoe gaat de werkdag er uitzien qua tijdsindeling? Het zou vervelend zijn als de som van handmatige transakties en beeldscherm transakties de acht uur per dag zou overschrijden! Met andere woorden, met behulp van dialoogsimulatie wordt de manier van werken duidelijk voor de gebruiker, met Transaktie analyse krijgt hij inzicht in zijn tijdsbesteding per dag. Daarmee is weer een van de sociale aspekten van de automatisering konkreet geworden. Een ander aspekt is het maximum aantal terminaluren dat een funktionaris met een beeldscherm mag werken. Regelmatig verschijnen in diverse landen berichten over oogklachten. Het gaat dan om werknemers die te lang per dag achter het beeldscherm zitten. Het is mogelijk dat hiervoor wettelijke regels komen. Probleem bij de vaststelling van het maximum is dan wel de soort transaktie. Immers, bij eenvoudige data-entry kijkt een datatypiste hoogstens naar het scherm wanneer ze een fout maakt. Iets dergelijks geldt ook voor tekstverwerking. Bij andere transakties is de kommunikatie met het systeem via het scherm veel intensiever. Hoe het ook zij, met behulp van de resultaten van Transaktie analyse kunnen de konsekwenties van dit soort voorschriften berekend worden. Het kan betekenen dat er meer beeldschermen moeten worden opgesteld dan strikt noodzakelijk is. Er kan nu bepaald worden wat de kosten van de voorschriften zijn. Daarbij gaat het niet alleen om de prijs van extra terminals, maar ook om de extra belasting die kan ontstaan wanneer de extra terminals tegelijk met de overige in gebruik zijn.

- Synchronisatie tussen beeldschermen en printers.

Vaak worden beeldschermen gebruikt in kombinatie met printers. Het kan in de vorm van een hardcopy-printer die een kopie van het beeldscherm afdrukt of in de vorm van een door een printapplikatie bestuurde printer. Het is vaak van belang in een vroegtijdig stadium iets te weten over de benodigde printkapaciteit. Zeker wanneer verscheidene beeldschermgebruikers een printer moeten delen. De benodigde printkapaciteit hangt af van de snelheid waarmee transakties op het beeldscherm kunnen worden uitgevoerd. Synchronisatie tussen beeldschermen en printers wordt mogelijk met de resultaten van Transaktie analyse. Aangezien de printsnelheid een parameter is in het rekenmodel, is eenvoudig het effekt van een snellere printer te bepalen.

- De belasting op het systeem.

Het is een bekend probleem dat er bij interaktieve toepassingen moeilijk iets valt te zeggen over de belasting en daarmee over de performance van het systeem. In een batch-omgeving zijn vergelijkende tests vrij eenvoudig uit te voeren. Een aantal testjobs

wordt op de diverse machines gedraaid en de doorlooptijden woren vergeleken. Bij de keuze van computersystemen gebeurt er nog steeds bij veel bedrijven het volgende.
De direktie besluit tot automatisering over te gaan. Er worden bij een aantal leveranciers offertes aangevraagd. Aangeboden systemen worden met elkaar vergeleken op basis van zaken als aantal instrukties per seconde (MIPS of KOPS), het aantal schijven dat kan worden aangesloten, software pakketten ter ondersteuning van de ontwikkeling van applikaties. Soms komt er nog iemand op het idee om te vragen naar responsetijden. Die moeten gemiddeld toch wel onder de 2 seconden liggen! De computerleverancier kan daar uiteraard geen uitsluitsel over geven, zolang de applikaties nog ontworpen moeten worden. Hij informeert vervolgens naar de soort toepassing en biedt een demonstratie aan bij een bedrijf waar een soortgelijke toepassing op een soortgelijk systeem draait met een vergelijkbaar aantal beeldschermen(!). Soms wil de vasthoudende potentiele koper toch graag zoiets als garantie hebben over de performance van het systeem waar hij de eerst komende vijf jaar aan vastzit. Ook voor dit soort hardnekkige onderzoekers hebben sommige computerleveranciers een effektief antwoord. Er wordt een presentatie gehouden over de aspekten die allemaal een rol spelen in de performance, zeg maar de responsetijd van een computer. Om daar toch enige lijn in te brengen heeft men een aantal metingen gedaan en die vastgelegd in grafieken. In die grafieken is bijvoorbeeld de responsetijd af te lezen bij een bepaalde konfiguratie, bij een bepaalde "zwaarte" van applikaties bij een bepaald aantal terminals en een aantal ENTERS per uur. Als de hardnekkige klant nu even aangeeft wat de zwaarte van zijn applikaties is, en om hoeveel "ENTERS" per uur het gaat dan kan er best iets over responsetijden gezegd worden. Op dat moment haakt ook deze klant af. En daarmee is het pleit gewonnen. Afsluitend wordt er nog iets gezegd over de uitbreidbaarheid van het systeem en over de onmogelijkheid om in dit vak vooruit te denken. Peinzend grijpt de direkteur naar zijn vulpen en tekent het reeds geheel ingevulde leveringskontrakt. Na verloop van tijd wordt het systeem geleverd en in bedrijf genomen. En dan nog begrijpt niemand hoe het komt dat een systeem, dat nog niet voor de helft in gebruik is, al helemaal vol zit. Maar ja, achteraf praten is ook veel eenvoudiger dan vooruitdenken. Met een methode als Transaktie analyse wordt in ieder geval al het mogelijke gedaan aan dat vooruitdenken. Transaktie analyse levert namelijk precies die cijfers op die de computerleverancier vroeg tijdens de presentatie over performance bij interaktieve toepassingen!
In een omgeving waar op een groot mainframe reeds talloze toepassingen draaien kan men zich afvragen wat het nut is van systeembelastingscijfers van een paar nieuwe transakties. Die cijfers

zijn niet direkt bruikbaar. Als men echter vastbesloten is performancebeheer konsekwent inhoud te geven, zouden van alle nieuwe transakties naast de andere resultaten van Transaktie analyse, ook de belastingscijfers moeten worden vastgelegd: al die reeds bestaande toepassingen hebben niet het eeuwige leven.
Het tweede soort resultaten van Transaktie analyse heeft te maken met de netwerkkant van het beeldscherm. Het gaat daarbij om twee zaken: het verkeer en responsetijden. Het verkeer is het aantal tekens per tijdseenheid. Aangezien het verkeer in de vorm van berichten of boodschappen wordt uitgevoerd is het beter te spreken over verkeer in termen van aantal berichten per tijdseenheid en de lengte van de berichten. Het is opvallend hoeveel literatuur er bestaat over netwerksimulatie op basis van verkeer en hoe weinig over de bepaling van dat verkeer uitgaande van transaktieontwerp, dialoogontwerp, beeldschermlay-out en soort terminal.
Transaktie analyse levert als resultaat per transaktie: de berichtlengte heen en terug, alsmede de tijd tussen twee berichten. Daarmee kan elke netwerksituatie verder worden berekend. Alternatieven voor transaktie-ontwerp of dialoogontwerp kunnen nu snel beoordeeld worden op konsekwenties voor het verkeer. Transaktie analyse is geen methode om op wonderbaarlijke wijze responsetijden mee te kunnen berekenen in een situatie waarin het netwerk en de computer nog gekozen moeten worden.
Transaktie analyse is een begrotingsmethode die de cijfers levert voor de sociale aspekten van de automatisering en voor het verkeer en de systeembelasting.

36.2 Transaktie analyse in grote lijnen

Startdokument voor Transaktie analyse is het transaktieschema, zie Fig. 36.1. Dat wordt door informatie-analisten opgesteld in samenwerking met gebruikers, soms maken gebruikers ze zelf. Dat transaktieschema bevat de kwalitatieve beschrijving van de procedure aan het beeldscherm: welke handelingen er worden verricht, wanneer er op de ENTER-toets wordt gedrukt en wat het systeem doet. Daarmee zijn de volgende aspekten van de transaktie vastgelegd: de menselijke handelingen, de dialoog, het verkeer tussen beeldschermen en computer en de verwerking door het systeem. Dat schema bereikt zijn definitieve vorm via dialoogsimulatie en wordt vertaald naar een detailschema. Op het detailschema komt alles van het transaktieschema weer voor, maar nu in de vorm van standaard parameters. "Het intypen van het van een klantnummer" verschijnt nu onder parameterkode 1, met daarbij het aantal aanslagen. Zo wordt de hele transaktie via parameters gekwantificeerd. Het detailschema is input voor een rekenprogramma. Dat rekenprogramma levert altijd drie pagina's output: het parameter-

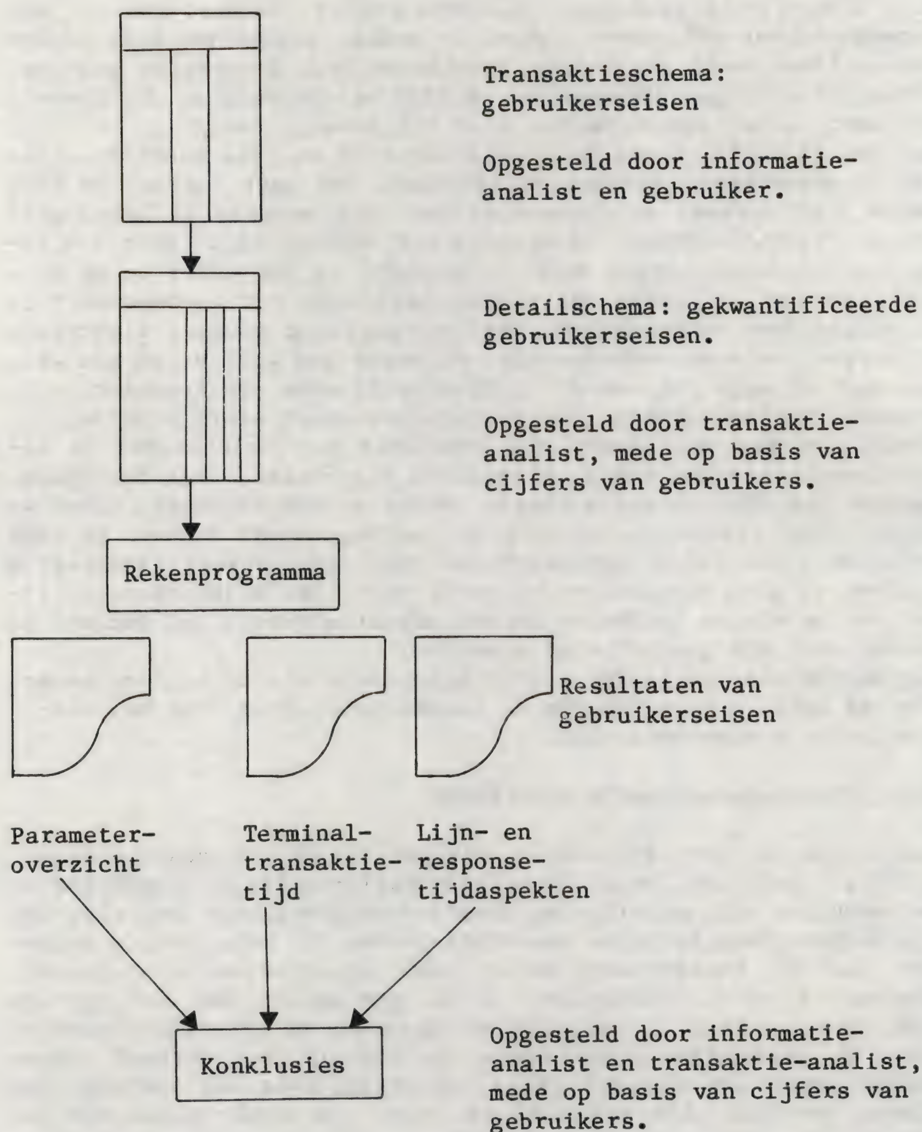


Fig. 36.1 Transaktie analyse in beeld.

overzicht met de totalen van alle parameters, de pagina Terminaltransaktietijd met de tijd die een transaktie duurt inklusief de samenstelling van die tijd en de pagina Lijn- en responsetijdaspekten, die gegevens bevat over het verkeer en de verwachte responsetijden. Deze laatste pagina is niet van belang als het gaat om een ergonomische Transaktie analyse. Bij die analyse gaat het erom gebruikers gegevens te verstrekken over het aantal beeldschermen, het aantal beeldschermuren enzovoort. Op het detailschema worden dan alle parameters die te maken hebben met de kommunikatie tussen beeldscherm en computer weggelaten, evenals de parameters die de verwerking door de programma's betreffen. De relevante parameters betreffen dus de procedure aan het beeldscherm. Die parameters behoren eigenlijk tot het terrein van de informatie-analist en worden daarom ook beschreven in het deel dat voor hen bedoeld is. Zij zouden dan het detailschema moeten kunnen maken en aanleveren bij de transaktie-analist of zelf het rekenprogramma kunnen starten. Met de verwerkingsparameters en verkeerparameters kan tijdens het logisch ontwerp de verwerking alvast in grote lijnen in kaart worden gebracht. Extreme responsetijden komen nu al boven water en kunnen een bijdrage leveren in de toegangspadanalyse.
Om de verwerkings- en verkeersparameters in het detailschema te kunnen opnemen is enige technische kennis omtrent screenmanagement, beeldschermbesturing en verwerking door applikatieprogramma's en database-managementsystemen van belang. De transaktieanalist wordt geacht die kennis te bezitten of te verwerven. Na een kursus als (17) weet hij in ieder geval wat er nodig is om een detailschema volledig in te vullen.
De drie pagina's tezamen leveren twee soorten resultaten op:
- ergonomische, die voor de gebruikers van belang zijn, onder andere als terugkoppeling op de cijfers die ze hebben verstrekt,
- technische, ten aanzien van verkeer, responsetijden en systeembelasting. Tijdens het logisch ontwerp gaat het om indikaties, verwachte technische resultaten, tijdens het technisch ontwerp om konkrete cijfers.

Soms kunnen globale cijfers al van groot belang zijn voor het doorgaan van een projekt. Tijdens het vooronderzoek kunnen globale transaktieschema's worden gemaakt die met name de ergonomische aspekten globaal in kaart brengen. Dan kan men een indruk krijgen van het benodigde aantal beeldschermen, het aantal uren dat er transakties moeten worden uitgevoerd. Transaktie analyse is een methode waarbij volgens een vaste procedure attributen van het entiteitstype Transaktie worden bepaald. Het rekenprogramma dat bij de methode hoort, wordt gratis verstrekt aan deelnemers van (17).

36.3 Kwantiteiten

Hoewel automatisering tot voor kort hoofdzakelijk een kwestie van logika was, begint kwantificering een steeds belangrijker plaats in te nemen.
Heel duidelijk is dat bij het database-ontwerp. Het is meestal op zich al een prestatie om een goede database-struktuur te ontwerpen. Veel databases die een uitstekende struktuur hebben, blijken echter een geweldige rem te zijn op de performance van het totale systeem. In de praktijk wordt dan met allerlei tuningshulpmiddelen naar de optimale situatie gezocht. Steeds vaker wordt er een scheiding aangebracht tussen logisch database-ontwerp en fysiek database-ontwerp. Het logisch database-ontwerp is een zaak van systeemontwerpers, aktief tijdens de fase logisch ontwerp. Het fysiek ontwerp is een zaak van database-specialisten, tijdens de fase technisch ontwerp.
Zij vertalen het logisch ontwerp naar het fysieke ontwerp zoals dat kan werken met het database-managementsysteem op de betreffende computer. Bij dat fysieke ontwerp is het gebruik van de gegevens en de frekwentie van het gebruik van groot belang. De toegangspaden tot de gegevens kunnen worden toegesneden op het gebruik.
In sommige systeemontwikkelingsmethoden wordt dan ook konsekwent en systematisch in kaart gebracht hoe vaak gegevens benaderd worden en via welk zoekargument. Met andere woorden, kwantitatieve gegevens over gebruik, pieken, frekwenties zijn daar van groot belang. Een goede specifikatie van dit soort gegevens maakt een goed database-ontwerp mogelijk. Voor netwerken en netwerkontwerp geldt hetzelfde. Basisgegeven voor een netwerk is het verkeer. Verkeer betekent het aantal tekens per tijdseenheid. Gegevens over verkeer zijn altijd afgeleid van gebruik van gegevens, frekwenties en pieken.
Transaktie analyse maakt een systematische vertaling mogelijk van gebruikerskwantiteiten naar verkeer, aantal beeldschermen, belasting van het systeem. Het is dus de moeite waard om te zorgen voor zo nauwkeurig mogelijke gegevens. Op dit terrein blijken de informatie-analisten, transaktie-analisten en database-ontwerpers vaak te gemakkelijk door de knieen te gaan voor het schouderophalen van de gebruikers.
Dat zullen dan wel vaak dezelfde gebruikers zijn als degenen die later klagen over slechte performance van het systeem. Van het taktisch management mag dan ook verwacht worden dat het zijn invloed aanwendt zowel in de richting van de gebruiker als in die van het uitvoerende automatiseringsmanagement en zijn medewerkers.
De gebruikers moeten geattendeerd worden op het belang van goede

gegevens, hun eigen belang wel te verstaan.
Aan uitvoerende automatiseringsmedewerkers moet worden duidelijk gemaakt dat er gekozen is voor bepaalde ontwerpmethoden en dat daarvan niet wordt afgeweken. In het geval van Transaktie analyse bijvoorbeeld worden dan de resultaten en de konklusies terugvertaald naar de konsekwenties voor de gebruikers.
Dat betekent dat gebruikers tijdens het logisch ontwerp al te horen krijgen tot welke situaties hun cijfers leiden. Konklusies in cijfers maken de meeste gebruikers bereid nog eens na te denken over de door hen verstrekte gegevens.
In de kommunikatie tussen taktisch automatiseringsmanagement en taktisch gebruikersmanagement moet dit leiden tot een oplossing. Meestal is er gebrek aan tijd in de gebruikerswereld om bepaalde kwantiteiten te onderzoeken, maar op het niveau van taktisch management dient dit soort zaken geregeld te kunnen worden.
Dat niet alle situaties even simpel zijn, moge blijken uit het volgende voorbeeld.
Orderentry kan eenvoudig klinken. Er bestaan echter bedrijven waar men zeer uitgebreide, ingewikkelde orders verwerkt. Een voorbeeld daarvan zijn de leveranciers van verwarmings- en airconditioningsinstallaties. Er wordt een order ontvangen voor de aangeboden installaties. Een airconditioningsinstallatie bestaat uit een aantal eenheden. Eenheden voor verwarming, bevochtiging, naverwarming, ventilatie, koeling enzovoort. Iedere eenheid op zich bestaat weer uit een aantal eenheden, elk met hun technische specifikaties. Een gebruiker kan best een morgen bezig zijn met het invoeren van een order. Omdat iedere installatie bestaat uit een kombinatie van elementen zal al snel duidelijk zijn dat er per element een transaktie zal worden ontworpen. Het maken van transaktieschema's en het uitvoeren van dialoogsimulatie blijft dus zinvol. Of het uitvoeren van Transaktie analyse ook zinvol is, is de vraag. Het repeterende karakter is nihil, kwantiteiten zijn moeilijk aan te geven, het gaat vaak om een zo klein aantal orders per jaar dat per order een feest gevierd kan worden, na dialoogsimulatie heeft de gebruiker een indruk van de tijdsbesteding en de verwerking zal bestaan uit korte pieken. Het is moeilijk een algemene regel te geven, maar als Transaktie analyse wordt uitgevoerd moet het doel ervan goed duidelijk zijn. De informatie-analist hoeft alleen maar de ergonomische resultaten van Transaktie analyse te kennen en dus moet hij zich afvragen of een dergelijke analyse voor hem en de gebruikers iets oplevert. De andere resultaten van Transaktie analyse zijn ter beoordeling aan de transaktie-analist. In CxN-omgevingen kunnen er best technische redenen zijn om Transaktie analyse uit te voeren. In het deel voor de transaktie-analist zal behandeld worden bij welke toepassingen en omgevingen Transaktie analyse zinvol is.

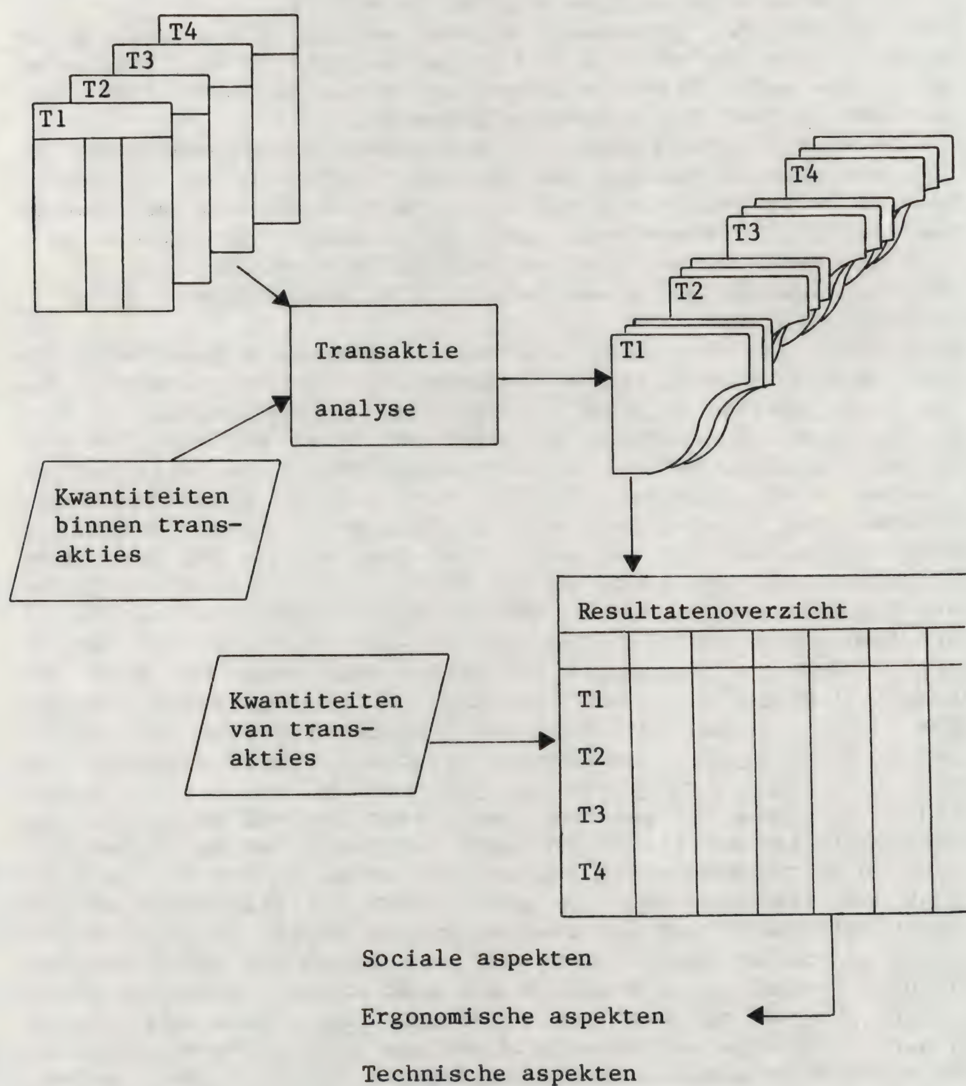


Fig. 36.2 Verwerking van resultaten

Samenvattend kan gesteld worden dat het belang van kwantiteiten in z`n algemeenheid iedereen duidelijk moet zijn. Ook de beste ontwerpmethode staat of valt met goede gebruikersgegevens. Met name bij het onderwerp kwantiteiten is het een taak van het taktisch management te zorgen voor konsekwente toepassing van de gekozen methoden en het scheppen van de mogelijkheden voor de uitvoering. Wanneer een aantal zaken moeilijk kwantificeerbaar is mag dat nog geen reden zijn om dan helemaal maar niet te kwantificeren. Richtgetallen zijn beter dan geen getallen.
Maar zelfs als om allerlei redenen een goede kommunikatie met gebruikers niet mogelijk is, kan de gebruikers konstant worden voorgerekend tot welke konsekwenties aannames van de automatiseerders leiden. Dat is het voordeel van een exakte rekenmethode als Transaktie analyse.
Het transaktieschema is de start van dialoogsimulatie. Het definitieve transaktieschema is startdokument voor Transaktie analyse. Uiteraard kan een transaktieschema uit een aantal bladen bestaan, evenals het detailschema. Het rekenprogramma levert voor iedere transaktie drie pagina`s met resultaten. In een projekt gaat het meestal om enkele tientallen transakties. In Fig. 36.2 is de manier van werken weergegeven. De vertaling van de transaktieschema`s naar detailschema`s en de uitvoering van het rekenprogramma zijn nu samengevat in het vakje Transaktie analyse. Bij kwantiteiten binnen transakties gaat het bijvoorbeeld om het aantal orderregels per order, bij kwantiteiten van transakties over het aantal orders. Van iedere set van drie pagina`s worden de benodigde resultaten overgenomen op een resultatenoverzicht. Als het gaat om een ergonomische Transaktie analyse dan worden per Transaktie analyse alleen die resultaten overgenomen die van belang zijn voor de konklusies. Een gedetailleerde uitwerking is te vinden in het deel voor de infrormatie-analisten en de transaktie-analisten.

Hoofdstuk 37
Netwerkontwerp

37.1 Wat zijn netwerken?

In de nederland- en engelstalige handboeken over datakommunikatie, computernetwerken staan meestal uitstekende verhalen over netwerkstrukturen, netwerkapparatuur en voorbeelden van netwerken. We zullen die onderwerpen hier niet herhalen, maar alleen de verschillende vormen van netwerken noemen om de relatie met het ontwerpen ervan aan te geven. We laten daarbij de local area networks (LAN`s) even buiten beschouwing.
Het gaat dus om wide area networks (WAN`s). Dat zijn netwerken die dienen om geografisch gescheiden lokaties met behulp van datakommunikatielijnen te verbinden. Een verbinding tussen een cluster van beeldschermen en een centrale computer is in die zin een netwerk. In de praktijk doen zich bij een dergelijke simpele verbinding reeds grote problemen voor. Soms worden toepassingen die zonder te denken aan een netwerk zijn ontworpen, ineens ook aangeboden aan remote gebruikers via een geschakelde lijn van 1200 BPS. Dan blijken de responsetijden die flitsend zijn voor lokale gebruikers ineens te zijn opgelopen tot zes seconden! Een ander bekend probleem is de kombinatie van beeldschermen en printers via een lijn verbonden met een computer: als er geprint wordt kunnen de beeldschermgebruikers het verder wel vergeten. Meestal komen deze problemen pas naar voren bij de in bedrijfstelling. Over netwerk-ontwerp gesproken!

Dezelfde problematiek, alleen in veel sterkere mate, komt naar voren bij netwerken met een boomstruktuur. Daarbij wordt in stappen het verkeer van een groot aantal lokaties gekoncentreerd naar steeds minder lijnen met een steeds hogere kapaciteit. Ook hier kunnen de lijnen de knelpunten worden, maar ook de koncentrators die de berichtenstromen samenvoegen, kunnen maar een bepaalde hoeveelheid verkeer aan.
Maasvormige netwerken met alternatieve routes komen in de benelux nauwelijks voor. In de literatuur wordt, om een voorbeeld te vinden, dan ook bijna altijd teruggegrepen naar amerikaanse netten als ARPA.
De meest voorkomende alternatieve route naast de vaste lijn is de geschakelde lijn. In het algemeen is de maximumsnelheid van een geschakelde lijn lager dan die van een vaste lijn. Bij het netwerkontwerp hoort ook de bepaling van de performance in noodsituaties. In noodsituaties hoeven echter meestal niet alle toepassingen in de lucht te blijven. Als bekend is welke transakties in noodsituaties moeten blijven draaien en Transaktie analyse is uitgevoerd, is het niet moeilijk de benodigde kapaciteit te bepalen. Dat geldt uiteraard evengoed voor netwerken ten dienste van computeruitwijk. Bij maasvormige netwerken gaat het om netwerken die bestaan uit lijnen en netwerkcomputers die niets anders doen dan het verkeer regelen. Er zijn leveranciers die dergelijke systemen kant en klaar leveren. Als een bedrijf dat beschikt over een groot aantal computers op verschillende lokaties, besluit een aantal van die computers met elkaar te verbinden, dan kan zo'n netwerk topologisch gezien wel op een maasnetwerk lijken maar in feite is het een verzameling computer-computerverbindingen. De meeste operating systems, electronic mailsystems en applikaties zijn niet geschikt om routeringsfunkties uit te voeren. In principe zijn het funktioneel gezien, bijna altijd ster- of boomvormige netwerken. In het netwerk is een systeem als centrale host aangewezen. Peer to peer kommunikatie maakt gebruik van de onderste verbindingen in de boom, maar maken van de struktuur nog geen maasvorm. Wat dit betreft beperken we ons dus tot koppelingen tussen computers ten dienste van interaktieve toepassingen. Voor de bepaling van de netwerkkapaciteit voor batchverkeer moet men beschikken over wat datakommunikatiekennis, de recordgrootte en het aantal records per batch. Dat dit al problemen kan opleveren blijkt uit de micro-mainframe-koppelingen met onverwacht lange transporttijden.
Soms leveren computerleveranciers een netwerkarchitektuur. Dat betekent dat het berichtenverkeer via standaard netwerkfunkties wordt afgehandeld en dus dat er aan het bericht van het beeldscherm of de applikatie altijd een hoeveelheid besturingstekens wordt toegevoegd om het transporteerbaar te maken. Transaktie

analyse brengt de berichtlengte in beeld. Afhankelijk van het soort netwerk moet daaraan worden toegevoegd de overhead van lijn of netwerkprocedures. Het is duidelijk dat de geografische situatie de basis is voor het netwerkontwerp. Laten we even aannemen dat het gaat om een sternetwerk. In het hoofdkantoor van een bedrijf staat een mainframe, op een twintigtal vestigingen worden clusters van beeldschermen geplaatst. Van iedere vestiging wordt een lijn uitgerold naar het mainframe en de ster is gerealiseerd. De eerste vraag is nu, welke kapaciteit een lijn moet hebben om een bepaald aantal beeldschermen te kunnen bedienen. Als blijkt dat de kapaciteit van de lijnen naar twee naast elkaar gelegen vestigingen maar voor een klein deel worden gebruikt, is de tweede vraag of die lijnen niet door een of ander apparaat kunnen worden gekombineerd of gekoncentreerd tot een enkele lijn. Bij beide vragen gaat het om hetzelfde probleem: de bepaling van de benodigde kapaciteit. Lijnen zijn verkrijgbaar in een aantal kapaciteitsklassen, afhanlelijk van het aantal tekens dat per seconde kan worden getransporteerd. De vraag is dus steeds: welke kapaciteit is nodig om een aantal beeldschermen te bedienen of, omgekeerd, hoeveel beeldschermen kan een bepaalde lijn bedienen? Als er een schermlay-out van 500 tekens naar een beeldscherm is gestuurd, waar de gebruiker een minuut naar kijkt, kunnen er in die tijd heel wat schermlay-outs over dezelfde lijn naar andere beeldschermen van het cluster worden gestuurd. Kortom, of een lijn een aantal beeldschermen kan bedienen hangt van het transaktie-ontwerp af. Hoe minder vaak er ge-ENTER-ed wordt, hoe meer beeldschermen een bepaalde lijn kan bedienen! In algemene termen samengevat: het gaat om de berichtlengte van en naar de computer en de tijd tussen de berichten. Voor het netwerk maakt het in principe niet uit of een hoeveelheid verkeer gegenereerd wordt door een groot aantal beeldschermen met veel menselijke handelingen, dus weinig ENTER`s per tijdseenheid per beeldscherm of door een klein aantal beeldschermen dat in z`n geheel hetzelfde aantal ENTER`s per tijdseenheid levert. Vandaar dat in veel handboeken over netwerkontwerp de transaktie en het aantal beeldschermen niet aan de orde komen. Men gaat er gewoon van uit van de berichtlengte en tijdsverdeling gegeven zijn. Hoe die bepaald moeten worden als een transaktie is ontworpen, wordt in het midden gelaten. Dat is de witte vlek tussen systeemontwerp en netwerkontwerp.

37.2 Hoe worden netwerken ontworpen?

In de inleiding is al aangegeven dat er een witte vlek bestaat in de kommunikatie tussen systeemontwerpers en netwerkontwerpers. In het algemeen weten netwerkontwerpers precies welke gegevens ze

nodig hebben om de juiste apparatuur te kiezen. Per aansluiting aan het netwerk gaat het om de hoeveelheid verkeer. Verkeer is het aantal tekens per tijdseenheid of het aantal berichten van een bepaalde lengte per tijdseenheid. Het verkeer op een bepaalde aansluiting hangt af van het aantal beeldschermen, het dialoog-ontwerp, de schermlay-outs, het aantal ENTER's per tijdseenheid en het type terminal. Het aantal beeldschermen is soms bekend, het aantal ENTER's per tijdseenheid nooit, want dat hangt af van de menselijke handelingen aan het beeldscherm en die zijn nooit in kaart gebracht. Systeemontwerpers weten hoogstens wat er wordt ingetypt en wat het systeem moet displayen. Soms weet geen enkele specialist hoeveel tekens er naar een beeldscherm moeten worden gestuurd om een bepaalde lay-out op het scherm te zetten. Aangezien een leverancier van netwerkapparatuur niet betaald wordt om Transaktie analyse uit te voeren, gaat hij dus uit van wat ervaringscijfers en als het aan hem ligt, zal het netwerk niet over te weinig kapaciteit beschikken! In feite is het verhaal gelijk aan dat van de computerleveranciers. (33.2)
Kortom, het gaat zoals een verkoper van een value added network (VAN) in Londen eens zei: "We brengen het netwerk tot aan de voordeur, we vragen naar het aantal beeldschermen, het soort beeldscherm, de berichtlengtes en de plannen voor de komende drie jaar. Als ze dat allemaal niet weten geven wij een advies"!
En dat terwijl er zoveel handboeken bestaan over netwerkontwerp.

37.3 Netwerkontwerp in de vakliteratuur

Er zijn veel goede handboeken op het gebied van datakommunikatie en netwerkontwerp. Met name de kwalitatieve aspekten van netwerkontwerp zijn goed beschreven. Onderwerpen als gelaagde strukturen, het OSI-model, protocollen, packet switching, architekturen en strukturen komen in veel boeken voor. Het aantal boeken dat zich bezig houdt met de kwantitatieve aspekten van netwerken is al beduidend kleiner. Toch doet de inspanning die ervaren auteurs zich getroosten om de rekenprocessen uit te leggen, vermoeden dat de kwantitatieve aspekten van netwerk een wezenlijk onderdeel vormen van het ontwerp. Niet voor niets besteedt James Martin in (8) aparte hoofdstukken aan statistieken en wachtrijen. Het probleem is alleen dat in (8) de behandeling van formules en rekenprocessen wordt gevolgd door voorbeelden waarin de benodigde cijfers gegeven zijn. In de praktijk is juist het bepalen van die cijfers het probleem. Als voorbeeld nemen we de bepaling van het aantal terminals en het verkeer.
- Het aantal beeldschermen. In hoofdstuk 32 van (8) wordt het aantal terminals berekend uitgaande van de bezetting van de beeldschermen. De wachtrijtheorie die in een vorig hoofdstuk is

behandeld, speelt daarbij een belangrijke rol. In een voorbeeld (p.487) wordt de servicetijd bepaald aan de hand van de procedure aan het beeldscherm (p.488). Een primitieve vorm van Transaktie analyse dus, want hier wordt alleen de procedure van de gebruikers besproken. Bovendien gaat het om een voorbeeld en niet om een methode. Dan volgt de opmerking (p.489) "The correspondence between operators and terminals must clearly be an early decision in the calculation of numbers of terminals needed and must be based on study of the overall job of the person using the terminal." Precies, dat is transaktie-ontwerp ten voeten uit: tijdens het logisch ontwerp zowel kwalitatief (dialoogsimulatie) als kwantitatief (Transaktie analyse) de toekomstige situatie van de gebruiker samen met hem ontwerpen. Het gaat per slot van rekening om zijn "overall job".
- Het verkeer. Voorafgaand aan, en dus onafhankelijk van de bepaling van het aantal beeldschermen, wordt aangegeven hoe er gerekend moet worden met de hoeveelheid verkeer in het hoofdstuk Basic statistics. In het voorbeeld op p.400 is de verdeling van de berichtlengtes eenvoudig een gegeven. Op een klein zinnetje komt het aan: "A study of the man-machine dialogue which takes place has given the distributions of lengths shown in tables 30.3 and 30.4." Hoe die studie moet worden uitgevoerd en hoe daarin bijvoorbeeld het transaktie-ontwerp en het soort beeldscherm moet worden betrokken, blijft onbesproken. Diskussies met medewerkers van Martin over dit onderwerp, brengen hen duidelijk in verlegenheid.
Als de gegevens over het verkeer bekend zijn, bijvoorbeeld door uitvoering van Transaktie analyse, is het rekenwerk relatief eenvoudig. Daarom zullen we ons ook verder niet verdiepen in statistieken en wachtrijen, die al vaak genoeg zijn beschreven.
Het genoemde voorbeeld wordt in de hoofdstukken 36 tot 38 opgepakt om berekeningen uit te voeren ten aanzien van het aantal beeldschermen aan een lijn. Daarmee is nogmaals aangegeven hoe belangrijk al dat rekenwerk is ten aanzien van verkeer in netwerken, maar dat het gebaseerd is op gegevens, waarvan Martin verzuimt te vertellen hoe ze verzameld moeten worden, uitgaande van gebruikers die werken met beeldschermen. Tenslotte nog een voorbeeld van het gemak waarmee men over het bepalen van de verkeersgegevens heen stapt. Op p.695 staat een flow chart van een rekenprogramma CNDP van IBM dat begint met de simpele opdracht: "Determine no. of lines needed for each location." Ja, dat is nu net het probleem. De volgende opdracht luidt: "Set aside fully loaded lines." Dan moet dus eerst de "load" van alle lijnen worden bepaald. De lijnbelasting hangt enkel en alleen af van de hoeveelheid verkeer. James Martin heeft naast (8) vele boeken op zijn naam staan over dialoog-ontwerp, database design, distributed

processing en systeemontwikkeling. Bovengenoemde voorbeelden bewijzen dat ook bij Martin een witte vlek bestaat tussen systeemontwerp en netwerkontwerp.
Een ander boek op het gebied van computernetwerken is (18) waarin op uitstekende wijze de kwalitatieve aspekten van netwerken worden behandeld. De hoeveelheid rekenwerk in hoofdstuk 2 geeft weer aan hoe belangrijk de kwantitatieve aspekten in het netwerk-ontwerp zijn. Via rekenkundige modellen wordt de kapaciteit van netwerken bepaald. Ook hier is het verkeersaanbod een gegeven. "All you need to know is the mean input rate (...). For example, there are 3 packets/sec" (p.63), "let us now assume a mean packet size 1/u=800 bits/packet" (p.64), "An important question is how much traffic a network can support (...)" (p.65). Natuurlijk kan het belangrijk zijn de maximum kapaciteit te berekenen van een netwerk, maar nog belangrijker is de benodigde kapaciteit, gegeven een aantal beeldschermen, het soort beeldscherm en de transakties. Op p.67 blijkt het te gaan over de ontwerpfilosofie die is toegepast bij het ontwerp van het ARPANET. Vandaar dat in (18) het aantal beeldschermen en de toepassingen helemaal niet voorkomen. Het gaat om het ontwerpen van een "openbaar" netwerk dat berichten transporteert. Daarbij geldt maar een kriterium: de maximum hoeveelheid verkeer die wat het netwerk aan kan en de hoeveelheid die per aansluiting getransporteerd moet kunnen worden. Hoe dat verkeer ontstaat, is bij zo`n netwerk niet van belang. Dat soort netwerken wordt in het gemiddelde bedrijf niet dagelijks ontworpen. En zelfs als het gaat om het ontwerpen van een infrastruktuur, die wordt aangeboden aan een aantal bedrijven dan is naast het bepalen van de maximumkapaciteit, nog altijd de begroting van het werkelijke gebruik van nieuwe klanten onontbeerlijk. Al zou het alleen maar zijn om tijdens het ontwerp een inzicht te hebben in de vertraging van het netwerk en in de kosten, die afhankelijk zullen zijn van het gebruik en dus van de benodigde kapaciteit. Responsetijden en kosten zijn toch niet de onbelangrijkste aspekten van interaktieve toepassingen.
In (31) vinden we een uitstekende inleiding tot het datakommunikatiegebeuren. Er wordt een apart hoofdstuk gewijd aan responsetijden. Wat de kwantitatieve aspekten daarvan betreft is het (8) in het klein: iets over statistieken, iets over wachtrijen, een gedetailleerde kwalitatieve behandeling van alle elementen waaruit een responsetijd kan zijn opgebouwd en dan enige voorbeelden van berekeningen. "(...) een opstelling die al rekenende en schattende tot stand kwam en gold voor een bepaalde situatie" (p.154). "Het gebruik van de formules zullen we met een voorbeeld toelichten" (p.156). Er wordt een transmissieformule behandeld, waarin "i= het aantal informatietekens (...)." Nog minder dan in (8) wordt aangegeven hoe de cijfers in de voorbeelden te verkrijgen

zijn.
Heel praktisch lijkt (28), omdat het daarin gaat om de selektie van apparatuur voor netwerken. Op p.48 gaat het om "(...) identifying user needs". Het netwerkontwerp dient dus bij de gebruiker te beginnen en te resulteren in de hoeveelheid verkeer: "Study user needs to identify (...) traffic volumes and frequency (...). - Draw up a list of terminal characteristics and select possible terminal types (...) Draw up network configuration and select circuits" Last but not least: "Put it all together and make sure it works"(!)
Ten aanzien van de bepaling van de hoeveelheid verkeer wordt nog aangegeven hoe belangrijk dit gegeven is. "The basic parameters for the design of a data communications system are an estimate of the traffic it will have to carry and the frequency of use. Both average and peak values are essential". Uiteraard wordt nog beschreven hoe deze gegevens, die de basis vormen voor het netwerkontwerp, moeten worden verkregen. "For a new application some estimation will be necessary (...) the numbers should be over- rather than under-estimates". Dat is dus de aanpak.
Gelukkig wordt aan het eind van het boek een case study behandeld om de aanpak toe te lichten (p.194) "A 300 bps PSTN circuit will be adequate for the links Glasgow to Manchester, (...) the London to Manchaster link however will carry sufficient traffic to justify a leased line at 2400 bps". Dat is dus de praktijk. Geen enkele relatie met gebruikers, type terminals en hoeveelheid verkeer. De kunst is kennelijk gewoon in een keer de lijnsnelheid te schatten, zonder rekening te houden met alle eerder behandelde aspekten.
Soms lijkt het of in systeemontwikkelingsmethoden toch iets aan netwerkontwerp wordt gedaan. In (33) vinden we daar een voorbeeld van. Tijdens het funktioneel ontwerp (!) gebeurt er al het een en ander op het gebied van datakommunikatie (p.123). De specifikaties moeten bevatten: het aantal gewenste (telefoon-) lijnen, de aard van de telefoonlijnen en hun snelheid, het aantal en de soort modems, de plaats, het aantal en soort terminals, specifieke apparatuur als koncentrators, front-end-processors, etc. "Het specificeren van de eisen op dit terrein, de keuze en het beheer van het netwerk, dat dan ontstaat, wordt dan een vak apart ("netwerkbeheerder")." De vraag lijkt gerechtvaardigd, wat er nu eerst ontstaat: het netwerk of de netwerkbeheerder, als er al iets ontstaat tijdens het logisch ontwerp. Afgezien van de plaats van de paragraaf Data-communikatie, klopt de inhoud wel met de rest van (33): er wordt aangegeven wat er moet gebeuren, maar hoe, mag de lezer zelf bepalen. Ook hier wordt dus geen konkrete bijdrage geleverd tot het ontwerpen van een netwerk.
In (27) blijkt alleen deel 3 over computernetwerken te gaan, de

andere delen bevatten weer kwalitatieve verhandelingen over allerlei aspekten van datakommunikatie. Er worden vele soorten formules kort beschreven, slechts in een enkel geval met een rekenvoorbeeld erbij. Het is de vraag in hoeverre de formules praktisch bruikbaar zijn, in ieder geval ontbreekt ook hier elke indikatie over het verkrijgen van cijfers, gegeven een ontworpen transaktie, een dialoog en een type terminal.
Gezien de titel, wekt (32) toch ook enige verwachtingen ten aanzien van het netwerkontwerp. In verhouding met andere boeken worden hier vrij veel voorbeelden gegeven van eenvoudige netwerken. Helaas is alles volledig gericht op de situatie in de Verenigde Staten. Er wordt veel kwalitatieve informatie verstrekt, ten aanzien van de kwantitatieve aspekten zegt de schrijver: "A key concept followed throughout the book is that, if the user's equirements and previously sited design constants can be sufficiently quantified at the very first planning step, the design techniques and algorithms presented in subsequent chapters can be used to configure the minimum-cost network that will meet the stipulated performance criteria" (p.45). Deze opmerking zou in geen van bovengenoemde boeken misstaan hebben. Daarmee is de witte vlek tussen systeemontwerp en netwerkontwerp precies aangegeven: "(...) if the user's requirements (...) can be sufficiently quantified (...)".
Transaktie analyse begint bij het transaktieschema dat in samenwerking met gebruikers wordt opgesteld en eindigt met cijfers, die direkt kunnen worden gebruikt in de rekenvoorbeelden, de cases en de formules voor het netwerkontwerp in de verschillende handboeken. De informatie over netwerkontwerp van bovengenoemde vakliteratuur samenvattend kunnen we vaststellen dat de kwalitatieve verhandelingen meestal leerzaam zijn, er wordt aangegeven dat het uiteindelijk gaat om de kwantificering van gebruikers eisen, en soms worden voorbeelden van berekeningen gegeven. De witte vlek tussen gebruikerseisen en cijfers voor het ontwerp kan worden ingevuld met Transaktie analyse.

37.4 Netwerkontwerp en Transaktie analyse

In de analysefase zullen al enkele aspekten in kaart gebracht zijn ten aanzien van funkties, funktionarissen en de geografische situatie. Het resultaat van dialoogsimulatie is een set uitgekristalliseerde transaktieschema's. De rechterhelft van het transaktieschema bevat een aanduiding van de benodigde gegevens. Na de vergelijking met het gegevens- of datamodel is dus bekend welke gegevens per transaktie worden gebruikt. Toen dialoogsimulatie werd uitgevoerd is ook in kaart gebracht om welke gebruikers het ging. Daarmee is het verband tussen transakties en werk-

plekken bekend. Werkplekken zijn geografisch bepaald. Op die manier ontstaat de topografie van het gegevensgebruik.
De volgende fase is het opstellen van de topografie voor gegevensopslag. In veel bedrijven gaat het om een rijdende trein en zijn er al computers aanwezig. Daarmee is dus al een uitgangspunt aanwezig voor de gegevensopslag.
De kombinatie van beide topografiëen leidt tot een netwerktopografie. Wanneer verscheidene mogelijkheden voor gegevensopslag onderzocht moeten worden, omdat er nog keuzemogelijkheden bestaan ten aanzien van de lokatie van computersystemen, zullen er verscheidene netwerktopografiëen kunnen ontstaan. De resultaten van Transaktie analyse leveren per lokatie het aantal beeldschermen, de hoeveelheid verkeer en dus de benodigde kapaciteit van iedere netwerkaansluiting. Daarmee is te bepalen of, en zo ja, waar het zinvol is om apparatuur als multiplexers en koncentrators in het netwerk op te nemen. Vervolgens kan er op basis van de kosten per oplossing gekozen worden uit de ontworpen topologiëen. In deel 5, voor de transatie-analisten wordt dit verder uitgewerkt.
Voor netwerken met een centrale opslag van gegevens geldt in principe dezelfde aanpak, alleen is de netwerktopologie al gegeven: de ster. Of het zinvol of nodig is om van een ster over te gaan op een boomstruktuur met behulp van koncentrators, kan weer berekend worden aan de hand van de verkeersresultaten van Transaktie analyse.
Door toepassing van Transaktie analyse wordt de brug geslagen tussen cijfers die gebruikers kunnen verstrekken en die netwerkontwerpers nodig hebben.
Zoals is aangegeven in de paragraaf Transaktie analyse in grote lijnen, kan globale Transaktie analyse een globale indruk geven van aantallen beeldschermen en hoeveelheden verkeer en daarmee van een paar mogelijkheden voor het netwerk. Per projekt worden tijdens het logisch ontwerp en technisch ontwerp de gegevens steeds technischer en gedetailleerder. Het principe dat in Fig. 31.1 is weergegeven voor de computerkonfiguratie geldt ook voor het netwerkontwerp. Ook bij het netwerkontwerp gaat het erom de performance te beheren en niet achter de feiten aan te lopen tot gebruikers beginnen te klagen over lange responsetijden en niemand weet waar de knelpunten zitten.

37.5 Datanet 1 en Transaktie analyse

We zullen eerst enkele technische aspekten in kaart brengen die te maken hebben met omschakeling op Datanet 1.
Fig. 37.1 is een voorbeeld van veel voorkomende elementen in een bestaande situatie. We zullen nu een aantal mogelijkheden bespreken voor de omschakeling.

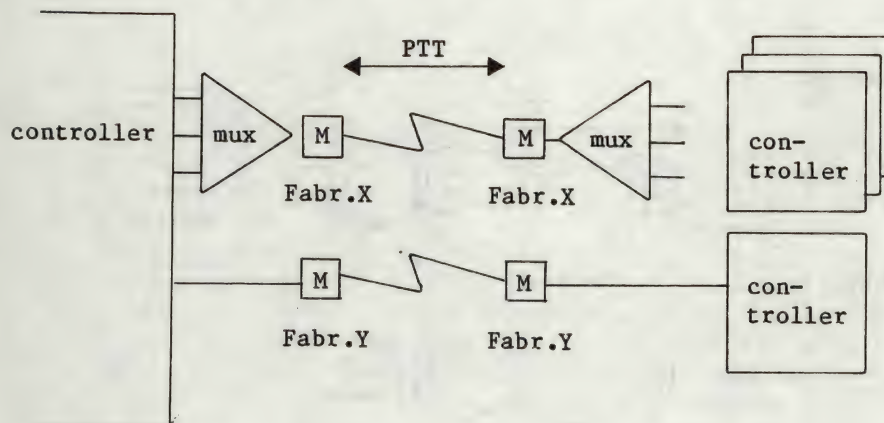


Fig. 37.1 Een bestaand netwerk

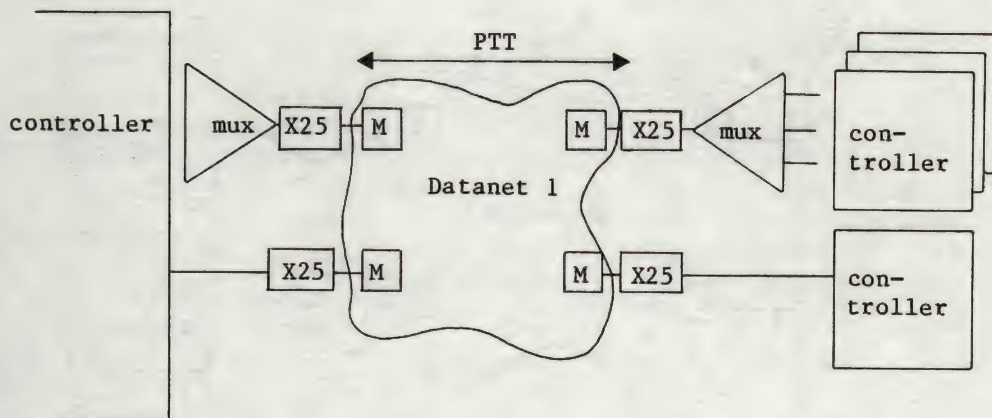


Fig. 37.2 Mogelijkheid 1.

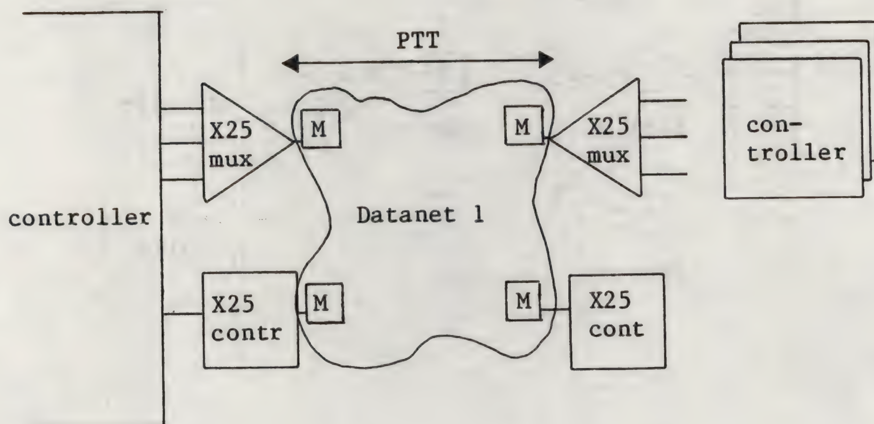


Fig. 37.3 Mogelijkheid 2.

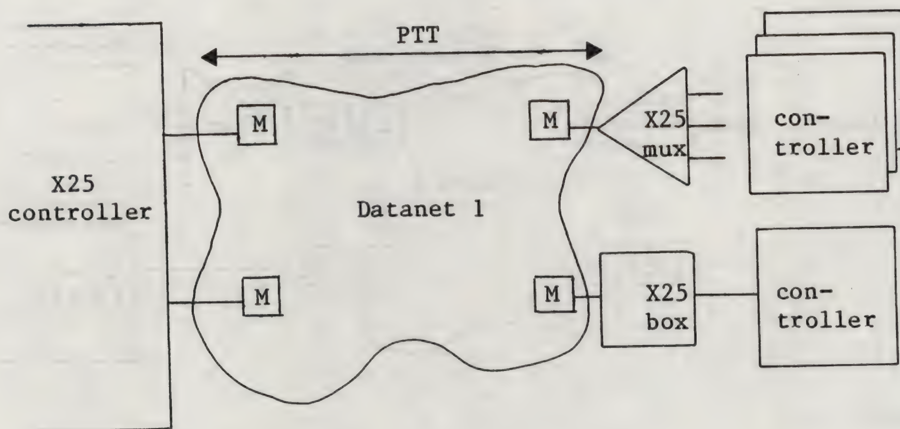


Fig. 37.4 Mogelijkheid 3.

- Mogelijkheid 1: Fig. 37.2
Hier gaat het om de omschakeling met behoud van zoveel mogelijk bestaande apparatuur. In grote lijnen komt het erop neer dat de modems van de PTT zijn, en de X25-dozen moeten worden toegevoegd.
- Mogelijkheid 2: Fig. 37.3
Bestaande multiplexers en concentrators worden vervangen door apparatuur die meteen X25 spreekt.
- Mogelijkheid 3: Fig. 37.4
Aan computerzijde vindt nu de multiplexing plaats door de X25-software/firmware. Aan de andere zijde van het net zal de apparatuur nog wel hetzelfde zijn als vroeger dus blijft een X25-black box nog nodig. Die black box moet wel afgestemd zijn op de X25-software/firmware aan hostzijde.
Een hardware spiegelbeeld als in de vorige situatie ontbreekt! Dat spiegelbeeld zit nu in de software van een leverancier die meestal een andere is dan de leverancier van de X25-box.
- Mogelijkheid 4: Fig. 37.5
Nu wordt aan terminalzijde gebruik gemaakt van de PTT PAD, een black box die wordt geplaatst tussen een X25-aansluiting en niet-X25-apparatuur. Die komt beschikbaar voor een beperkt aantal types terminals.
- Mogelijkheid 5: Fig. 37.6
Of deze oplossing mogelijk is hangt van het verkeer af. Daar de

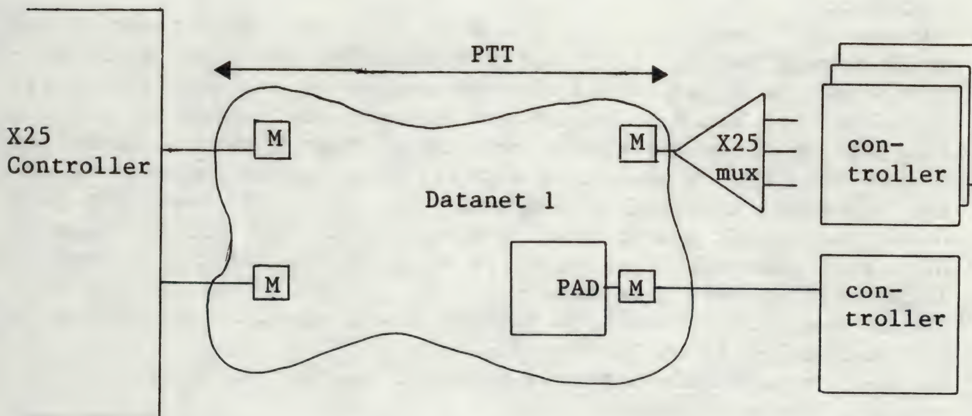


Fig. 37.5 Mogelijkheid 4.

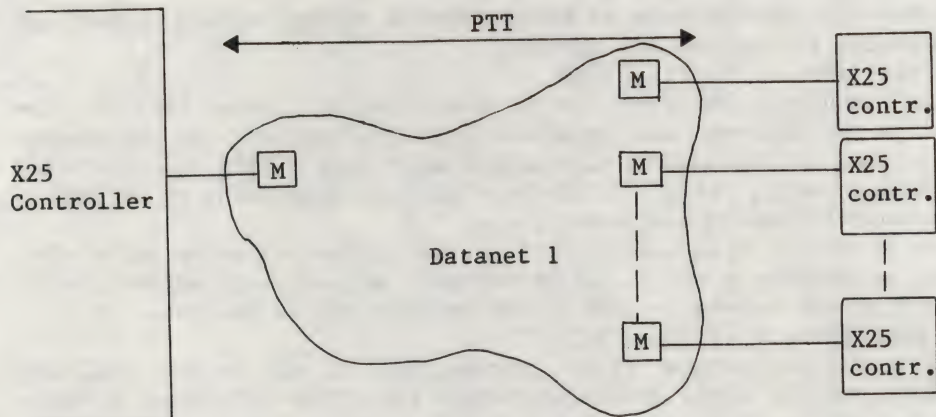


Fig. 37.6 Mogelijkheid 5.

lijnsnelheid aan hostzijde niet gelijk hoeft te zijn aan de lijnsnelheid aan terminalzijde is het mogelijk een groot aantal huidige lijnen te "koncentreren" tot een Datanet-1-aansluiting. In feite doet dan Datanet 1 de multiplexing.

- Kosten.

Wanneer men even afziet van andere aspekten van Datanet 1 dan geldt ruwweg dat de vaste kosten van van Datanet 1 per maand vergelijkbaar zijn met de lijnkosten per maand van een eigen netwerk. Daarbij komen de kosten van verkeer, die kunnen oplopen tot ca. 50% van de vaste kosten per maand. Mogelijkheid 1 leidt dan zonder meer tot de konklusie dat bij deze kosten nog eens de extra X25-dozen komen, die misschien over twee jaar overbodig zijn omdat dan mogelijkheid 3 of 4 gerealiseerd kan worden. Wanneer echter de bestaande apparatuur is afgeschreven, worden de mogelijkheden 3-5 weer interessant.

Op dat moment gaat ook een aantal andere aspekten van Datanet 1 een rol spelen in de besluitvorming.

Wanneer de bestaande situatie konstant zou blijven qua

- geografische situatie,
- aantallen aansluitingen,
- soorten gebruikers,
- leverbare faciliteiten,
- etc,

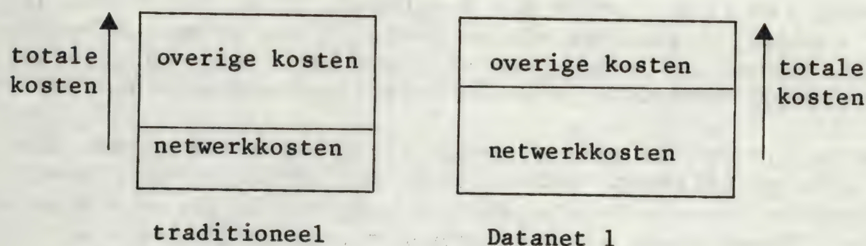


Fig. 37.7 Kostenverdeling.

dan zal er op basis van het bovengeschetste kostenplaatje nimmer een reden bestaan om over te gaan op Datanet 1!
Dit lijkt een kortzichtige en oppervlakkige manier van beoordelen. Van het totale kostenplaatje vormt het netwerk maar een klein gedeelte. In beeld gebracht kan de situatie zijn zoals in Fig. 37.7.
Daarbij vallen onder overige kosten met name de kosten voor personeel dat nodig is om het netwerk in de lucht te houden.
Tenslotte noemen we nog een aantal punten die bij de besluitvormingvan belang kunnen zijn:
- De PTT kent geen kortingssysteem voor de tarieven, maar een speciale behandeling voor een proefverbinding is bespreekbaar.
- Qua D.C.-apparatuur verandert er bij gebruikmaking van Datanet 1 in zoverre iets dat er in principe alleen een rek met standaard modems overblijft dat beheerd wordt door de PTT. Dit biedt toch enige voordelen ten aanzien van
  - beheer in het centrale rekencentrum en alle vestigingen,
  - scholing van mensen,
  - eenvoud van bediening.

Inventariseer het aantal aanwezige soorten apparatuur eens.
- Verhuizingen op de vestigingen verlopen veel eenvoudiger dan

bij eigen netwerken. Weliswaar duurt het bij Datanet 1 ook enige tijd voor een nieuwe aansluiting is gerealiseerd, maar dat is in te plannen. Men kan daarbij tegen geringe kosten beide aansluitingen elkaar laten overlappen. De verhuizingen van apparatuur betekenen dan niets anders dan de stekker op de ene plaats uit het PTT-modem trekken en elders weer in een andere PTT-modem prikken. Kortom dat aspekt van de installatie kan door een gebruiker gedaan worden, de modem en achterliggende zaken doet de PTT.

- Afhankelijk van de gewenste kapaciteit kunnen er nieuwe aansluitingen gerealiseerd worden, die aan hostzijde geen enkele konsekwentie hebben wat de apparatuur aangaat. En als de bestaande aansluitkapaciteit op een gegeven moment onvoldoende wordt, betekent dat een nieuwe aansluiting op een poort van de controller.
- In geval van storingen kan de PTT vanuit het beheerscentrum door de loopschakelaar op het modem de verbinding tot en met het modem testen.
- Blijkt er dan toch een niet verklaarbare storing over te blijven dan kan men een PDS-loop aanvragen. Dat wil zeggen dat de gebruiker pakketten het net instuurt, die geëchood worden ofwel door het pakket vanaf de andere zijde terug te sturen ofwel door een ACK-pakket terug te sturen. (ACK-pakketten zijn gratis)
- Bij af en toe optredende fouten kan men een TUC (transparant user communication facility) aanvragen. In dat geval gaan voor een bepaalde "verbinding" de pakketten via het beheercentrum en worden daar gelogged. Die file kan dan opgevraagd worden ter analysering.
- Heeft men nog geen aansluiting maar wil men toch testen, dan kan dat. Gedurende drie maanden stelt het beheerscentrum een 2400 bps modem beschikbaar voor een gekozen verbinding met het beheercentrum. Via die aansluiting kunnen bovenstaande funkties worden gerealiseerd.
- Wanneer in de bestaande situatie een lijnsnelheid onvoldoende blijkt te zijn, moet men de modems vervangen. Wat er met de overgebleven modems moet gebeuren is een vraag. Bij Datanet 1 bestelt men een aansluiting voor hogere snelheid voor 50% van de eenmalige aansluitkosten. Bijna uitsluitend een administratieve handeling dus.
- Uitwijksituaties worden nu eenvoudig. Als de X25-dozen bij de gebruikers voldoende gebruikersvriendelijk zijn, kan de hele omschakeling van terminals naar een uitwijkcomputer door gebruikers worden uitgevoerd. Datanet 1 is het operationele netwerk en het uitwijknetwerk tegelijk.
- Transaktie analyse.

Bij de overwegingen Datanet 1 als netwerk te kiezen zullen de

kosten een rol spelen. De maandelijkse kosten bestaan uit een vast deel en een deel dat afhangt van de hoeveelheid verkeer, uitgedrukt in segmenten van 64 bytes. Transaktie analyse berekent voor iedere transaktie de hoeveelheid verkeer in tekens per bericht heen en terug. Met deze gegevens is simpel uit te rekenen hoeveel segmenten er nodig zijn om die berichten te versturen. Per packet bedraagt het aantal segmenten 1 of 2. Aangezien ook de tijd tussen de berichten een resultaat van Transaktie analyse is, wordt nu ook het beheer van de kapaciteit van de aansluiting mogelijk.

37.6 De invoering van micro's

In het kader van dit boek zal het uiteraard met name gaan om de netwerkaspekten van microcomputers. Wanneer de invoering van micro`s is uitgevoerd zoals is aangegeven in het deel voor het strategisch management, is de problematiek te overzien, maar wanneer de gebruikers volledig vrij zijn gelaten in de keuze van hun micro, dan hebben we nu een enorm probleem. In de eerste plaats omdat dit soort aanschaffingen meestal zijn gedaan om een bepaald probleem op te lossen. Een evenwichtige behoeftenanalyse werd meestal vergeten, waardoor de aanwezige micro`s geen adequate invulling van de totale informatiebehoefte van hun eigenaars leve-

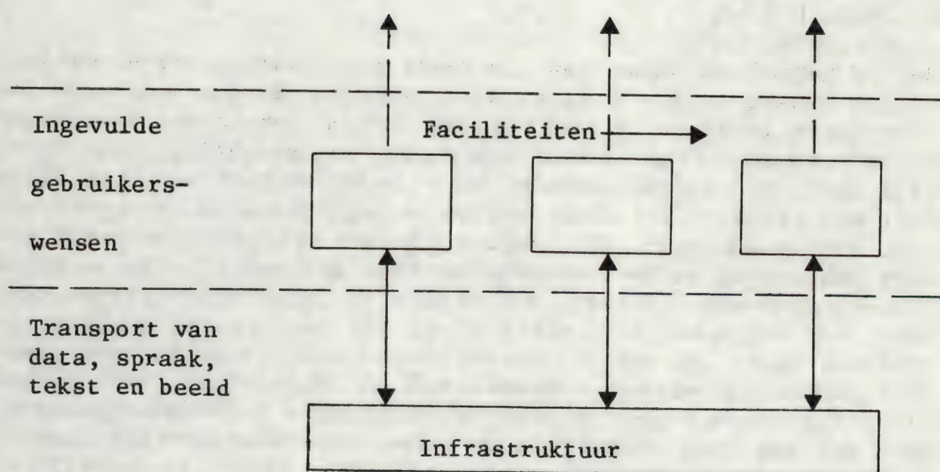


Fig. 37.8 Gebruikers en kantoorautomatisering

ren.
Vaak is er bewust of onbewust behoefte ontstaan aan oplossingen die nu eenmaal op een mainframe thuishoren.
Vroeg of laat komt de vraag naar een integrale aanpak van het automatiseringsgebeuren. Kort samengevat komt dat neer op een behoeftenanalyse per gebruiker, een faciliteitenontwerp en infrastruktuur om de faciliteiten te laten funktioneren. Werd vroeger met automatisering meestal de administratieve automatisering bedoeld, thans gaat het meestal over kantoorautomatisering. Dat betekent, dat niet alleen voor routinematige of rekenkundige bewerkingen faciliteiten worden gevraagd. Iedere medewerker op iedere werkplek kan gebaat zijn bij automatisering.
Als we vaststellen dat kantoorautomatisering niet hetzelfde is als tekstverwerking, dan moet kantoorautomatisering in de meeste bedrijven nog beginnen. Enerzijds zijn de meeste mensen nog niet toe aan het papierloze kantoor, anderzijds is het aanbod van bruikbare funkties nog erg beperkt. Het is daarom de hoogste tijd om alle behoeften in kaart te brengen. In de praktijk wordt daar pas aanbegonnen als het ineens moet, bijvoorbeeld omdat het mainframe aan vervanging toe is. Dan gebeurt alles onder tijdsdruk: er is geen goede analysemethode of de medewerkers zijn niet goed voorbereid.
Het gaat daarbij om konkrete gebieden als:
- electronic mail
- telex
- agendabeheer
- aktiviteitenlijst
- personal files
- archieffunkties

Het is bekend dat behoeften die vandaag in kaart gebracht worden, morgen anders kunnen liggen. Voor sommigen is dat een reden om maar niets in kaart te brengen. Het is ook bekend dat planningen in de automatisering vaak uitlopen. Dat mag natuurlijk geen reden zijn om niet meer te plannen. Nu zijn er talloze oorzaken waardoor een planning kan uitlopen. De belangrijkste is de onervarenheid van een planner. De ervaren planner weet dat problemen altijd onderschat worden, evenals de tijd die nodig is om ze op te lossen. Ervaring is terug te vinden in betrouwbare planningen, maar dat verandert natuurlijk niets aan onvoorziene externe invloeden. Zo is het ook bij behoeftenanalyses van gebruikers. Wanneer behoeften worden ingevuld met microcomputers, dan kunnen diezelfde micro's weer nieuwe onvoorspelbare behoeften oproepen, maar dat mag geen reden zijn om bebehoeften niet systematisch in kaart te brengen. Een eerste poging daartoe blijkt reeds heilzaam te werken ten aanzien van het nadenken over problemen en oplossingen: er komt nog altijd meer boven water dan zonder analyse.

Hoewel probleemanalyse buiten het kader van dit boek valt moge er toch op gewezen worden dat dialoogsimulatie een waardevol hulpmiddel kan zijn. Vooral aan de onervaren gebruiker kan heel snel en effektief de werking van een computer gedemonstreerd worden. De informatie-analist kan omdat hij enigzins bekend is met de materie, heel snel de simulator een praktijksituatie laten invullen, zodat de gebruiker zich gaat realiseren om welk soort oplossingen het gaat.
De volgende vraag die zich voordoet is de koppeling tussen micro`s. Stand alone micro`s zullen wel blijven bestaan, maar zijn niet van belang voor het netwerk. De meeste micro-eigenaars krijgen vroeg of laat behoefte aan kommunikatie met andere computers. Funktioneel kan men de kommunikatie binnen een bedrijf opsplitsen in kommunikatie tussen werkplekken onderling, tussen werkplek en afdelingsniveau en tussen werkplek en bedrijfsniveau. De diskussie over technische verschillen tussen mainframes en mini`s is gedoemd vast te lopen. Een praktische indeling is: dat het mainframe de centrale bedrijfscomputer is, mini`s de computers op vestigingen en op afdelingsniveau zijn en microcomputers op de werkplek staan.
Bij bedrijven waar slechts een minicomputer wordt aangeschaft beschikken we dus over een mainframe-mini. Een bedrijf waar alleen micro`s worden aangeschaft waarvan er een is voorzien van een harde schijf voor de "centrale" bestanden, beschikt over een micro-netwerk met daarin een mainframe-micro. Wanneer dat laatste bedrijf een vestiging van een groot bedrijf is, dan kan het zijn dat de mainframe-micro regelmatig voorzien wordt van update-bestanden door het mainframe. In principe kan het zelfs zo zijn dat zich tussen mainframe en mainframe-micro nog een minicomputer bevindt die gegevens distribueert naar diverse mainframe-micro`s en de werkplekmicro`s maken dan vervolgens weer gebruik van die gegevens. Immers, vroeg of laat komen gebruikers tot de ontdekking dat floppies een beperkte kapaciteit hebben, maar elke micro in het bedrijf uitrusten met een harde schijf is een kostbare aangelegenheid. Bovendien maken harde schijven de micro nu niet bepaald portable. Een bruikbare oplossing is een mainframe-micro in het netwerk met voldoende kapaciteit en kommunikatiemogelijkheid met de mini of met het mainframe. Dat brengt ons op een andere reden om te kiezen voor een mainframe-micro in een netwerk: de gateway funkties. Het is geen praktische oplossing iedere micro zelfstandig te laten kommuniceren met het mainframe. Terminal-emulatie is een aardige tussenoplossing, maar niet voor een netwerk van microcomputers: terminal-emulatie is immers alleen bruikbaar wanneer de micro ook de plaats in kan nemen van een terminal, hetzij via een datakommunikatieverbinding, hetzij via een aansluiting aan een cluster controller. Terminal-emulatie is

dus geschikt voor een stand alone micro. In een netwerk van met elkaar kommunicerende micro's dient er een de funktie van gateway te hebben, zeker wanneer de niveau's bedrijf, afdeling en werkplek funktioneel goed van elkaar gescheiden zijn. Dat kan bijvoorbeeld doordat op sommige werkplekken maar een klein deel van de afdelingsgegevens nodig is voor het goed funktioneren. De afdeling op zijn beurt heeft maar een klein deel van de bedrijfsgegevens nodig.

Bij netwerkontwerp gaat het altijd om twee zaken: de geografische situatie en het verkeer. Dat geldt ook voor netwerken voor micro's. Ervan uitgaande dat micro's werkplekgeörienteerd zijn, gaat het dus om lokale netwerken die zich in het algemeen binnen een gebouw of bedrijfsterrein bevinden. Zo worden werkplekken samengevoegd tot een funktioneel geheel. De term die in dat verband gebruikt wordt is "local area network " (LAN). Zo'n lokaal netwerk is een kabel die via de kabelgoten door het hele gebouw wordt geregen: bij iedere werkplek wordt een aansluiting gemaakt. Een aantal aspekten van zo'n lokaal netwerk is van belang. De enorme kapaciteit maakt het mogelijk alle verbindingen tussen werkplekken en computers in het hele gebouw te realiseren via die ene kabel.

Verder zijn alle aansluitpunten gelijkwaardig. Dat betekent dat interne verhuizingen geen konsekwenties meer hebben voor de bekabeling van het gebouw. Als op een werkplek met een bepaald beeldscherm een verbinding mogelijk is met een bepaalde computer, dan is dat op iedere andere werkplek met een aansluiting ook mogelijk.

Een computer is meestal verbonden met een aantal beeldschermen via een aantal kabels. Wanneer er verscheidene computers in gebruik zijn is iedere computer verbonden met zijn eigen beeldscherm. Het geheel levert een woud van kabels op in volgepropte kabelgoten. Dat geheel kan men wel een lokaal netwerk noemen, maar met een local area netwerk wordt bedoeld de vervanging van al die kabels door een "multi-purpose-kabel". In dit boek bedoelen we met een lokaal netwerk een local area netwerk (LAN). Met het bovenstaande is meteen aangegeven wat de funktie van zo'n lokaal netwerk is. Het is niet meer dan de vervanging van grote hoeveelheden kabels door een multi-purpose-kabel. Dat betekent tevens dat computers die via gewone kabels niet met elkaar kunnen kommuniceren dat ook niet kunnen via die multi-purpose-kabel. Hoewel in vele reklameplaatjes gesuggereerd wordt dat computers met computers kunnen worden verbonden, levert een lokaal netwerk niet meer dan een gewone kabel tussen twee computers. Twee computers van verschillend fabrikaat die niet met elkaar kunnen kommuniceren omdat de lijnprocedure verschilt, zullen ook via een lokaal netwerk over hetzelfde lijnprocedureblok struikelen. Voor

microcomputers geldt hetzelfde. In grote lijnen kunnen microcomputers voorlopig nog maar twee dingen: stand alone werken of via terminal-emulatie, fungeren als terminal aan een grotere computer. Kommunikatie tussen micro`s moet dus bijna altijd ontworpen worden, zeker voor administratieve toepassingen. Dat betekent dat de applikaties en de kommunikatie tussen de micro`s per micro ontworpen moeten worden. Daarbij speelt uiteraard een rol of er gekozen is voor een mainframe-micro of voor gelijkwaardige micro`s.
In dit soort netwerken zijn drie soorten verkeer te onderscheiden.

- Interaktief verkeer. Een micro heeft direkt een bepaald gegeven nodig van een andere micro.
- Interaktief batch-verkeer. Uitwisseling van bestanden waar de gebruiker op wacht, bijvoorbeeld een grijplijst voor de micro in het magazijn.
- Uitgesteld batch-verkeer. Dit is verkeer dat niet direkt nodig is voor het werken met de micro. Het zijn bestanden die bijvoorbeeld eenmaal per dag moeten worden uitgewisseld en in een batch-programma verwerkt worden.

Het interaktief batch-verkeer is karakteristiek voor micro`s. Een domme terminal heeft kontinu interaktief verkeer nodig met een computer. Een micro kan zo worden geprogrammeerd dat een grotere batch van gegevens kan worden geaksepteerd, die daarna post voor post interaktief wordt verwerkt.
Het zal duidelijk zijn dat het ontwerp van dit soort systemen een veelheid van varianten biedt. Tevens zal duidelijk zijn dat het bijzonder kostbaar wordt van een eenmaal gekozen koncept over te stappen op een ander koncept. De applikaties zijn dermate op elkaar afgestemd dat een wijziging aan de ene kant van het netwerk werk direkt konsekwenties heeft voor de andere kant. Voor alle duidelijkheid: het gaat hier om de bovenste laag van het bekende OSI-model. Wanneer er ontworpen is zonder gebruik te maken van in dat model ontwikkelde standaards, zijn wijzigingen in koncepten waarschijnlijk desastreus.
Maar afgezien van wijzigingen blijft het ook nog een probleem om te kiezen uit de alternatieven. Er zijn immers vele alternatieven te bedenken voor bijvoorbeeld de gegevensopslag.
Per manier van gegevensopslag is er een aantal mogelijkheden voor het transport van de gegevens. Dat wil zeggen dat de alternatieven goed gekwantificeerd naast elkaar gezet moeten worden. Dat betekent dat ook bij dit soort netwerken moet worden uitgegaan van gebruikerswensen en -kwantiteiten en dat die vervolgens moeten worden omgerekend naar konsekwenties voor de diverse alternatieven. Dialoogsimulatie en Transaktie analyse vormen ook hier de aangewezen weg om tot een goede kwantificering te komen.

Samengevat kunnen we vaststellen dat invoering van microcomputers zal leiden tot lokale netwerken. Local area networks die als "black box" op de markt worden gebracht voegen niets toe aan het gebruik van konventionele kabelverbindingen. Voor een netwerk van microcomputers is meer nodig dan kabels van welk soort dan ook. Dialoogsimulatie en Transaktie analyse vormen een uitstekende basis onder het ontwerp.

37.7 Van informatiebehoeften naar netwerkontwerp

Terugziend op de vele jaren van ontwerp van informatieverwerkende systemen moeten we vaststellen dat in veel bedrijven de systemen eerder ontstaan zijn dan ontworpen. Natuurlijk zijn programma's en databases ontworpen, maar het netwerk bijvoorbeeld is gewoon gegroeid. Centrale rekencentra hebben een konstant probleem met wildgroei op vestigingen en de erop volgende behoefte om de decentrale systemen toch te koppelen aan de bestanden op het mainframe. Van gestruktureerd ontwerp is dan geen sprake.
Het ziet er niet naar uit dat de informatiebehoefte van gebruikers zal afnemen. Velen van hen lezen de aan science fiction grenzende verhalen in sommige vakbladen en sluiten erop aan met hun wensen. De verdergaande opmars van micro's maakt het probleem niet eenvoudiger. Vroeg of laat moet de zaak onderling gekoppeld worden. Gebruikers gaan een keer de beperkingen van micro's inzien of de nadelen van computerbezit ervaren. In veel bedrijven is de automatisering een rijdende trein. De kosten zullen door de toenemende wensen van gebruikers alleen nog maar stijgen. Kortom, of het nu gaat om een vertrekkende of om een rijdende trein, beginnen met het opstellen van een goed informatieplan is een dringende noodzaak. Op de oude voet doorgaan gaat alleen maar meer geld kosten, terwijl de problemen voor gebruikers en automatiseerders alleen maar toenemen. Het informatieplan leidt uiteindelijk tot projekten die gerealiseerd moeten worden (35). In Fig. 37.9 is dat in grote lijnen aangegeven. De vier kolommen lopen van grof naar fijn.
Het zal duidelijk worden dat de vierde kolom iets van een heel andere aard weergeeft dan de andere drie. Het gaat echter om het koppelingspunt: funktionaris-procedures-werkplek-transakties in relatie tot de systeemontwikkeling.
De organisatiekolom begint bij het bedrijf en eindigt via de organisatiestruktuur bij de funktionaris, de werknemer. De bedrijfsfunkties worden verdeeld in aktiviteiten die steeds verder gedetailleerd worden, om te eindigen bij procedures. Een procedure is de manier van werken om een doel te bereiken of een deel van de taak te vervullen. Bij een funktionaris kan meer dan een procedure horen.

Organisatie	Funkties	Geografie	Automatisering
Bedrijf			Informatie-analyse
Divisies	Funktionele eenheden	Vestigingsplaats	Informatieplan
	Aktiviteiten		Prioriteiten
Afdelingen			
			Selektie
		Vestiging/ Kantoor	Automatiseringsplan
			GO/NO GO beslissing
Groep			
			Systeemontwikkeling
			Projektclusters
			Detailanalyse
			Logisch ontwerp
Funktionaris	Procedures	Werkplek	Transakties

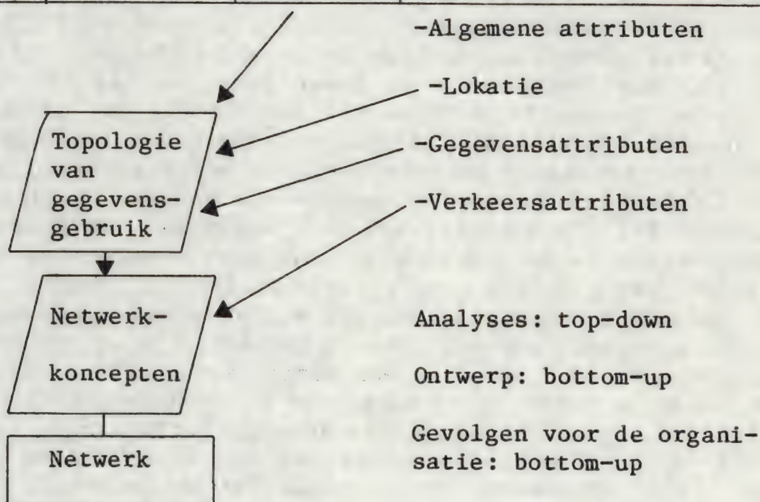


Fig. 37.9 Van informatiebehoeften naar netwerkontwerp.

De geografie van het bedrijf is het belangrijkste aspekt voor de topologie van het netwerk. Op het hoogste niveau in de kolom gaat het om de vestigingsplaats. In een stad of dorp kunnen zich verscheidene vestigingen bevinden. Iedere vestiging bestaat uit een aantal eenheden die onderling met een lokaal netwerk zouden kunnen zijn verbonden. Per geografische eenheid kan een aantal werkplekken worden onderscheiden. Hoe ingewikkeld de geografische struktuur ook is, uiteindelijk is iedere werkplek geografisch bepaald.
In de kolom automatisering is de aanpak beschreven die begint bij een informatieplan, waarvan projektclusters worden afgeleid. Binnen een cluster ontstaan transakties via transaktie-ontwerp per funktionaris, per werkplek. Op dat niveau kunnen alle organisatorische, personele, funktionele en sociale aspekten van de automatisering worden overzien.
Bij het analyseren van de informatiebehoeften zal men meestal niet afdalen tot het niveau van funktionarissen en procedures. Met andere woorden, vanuit de informatie-analyse kan men zich een lijn naar links voorstellen die per kolom op een andere hoogte ligt. De hoogte hangt helemaal af van het soort bedrijf, geografische situatie en de tijd die men wil besteden aan het informatieplan. Het is mogelijk een zeer gedetailleerd informatieplan op te stellen. In het deel voor informatie-analisten is aangegeven hoe transaktie-ontwerp wordt uitgevoerd tijdens het vooronderzoek. Dezelfde werkwijze kan worden toegepast tijdens het analyseren van de informatiebehoefte. Dat betekent dat het mogelijk is op werkplekniveau de zaak in kaart te brengen. Wil men tijdens de informatie-analyse niet zover gaan, dan is het mogelijk een globale Transaktie analyse uit te voeren. Dan ontstaan globale transaktieschema`s die tijdens het logisch ontwerp de basis vormen voor het transaktie-ontwerp. In beide gevallen ontstaan dus resultaten van Transaktie analyse en daarmee attributen voor de transakties. De lokaties van de transakties, de geografie van de werkplekken en de gegevens per transaktie maken het mogelijk het gebruik van gegevens geografisch in kaart te brengen. Als er maar een plaats is waar de gegevens kunnen worden opgeslagen, dan is nu de topologie van het netwerk bekend. Zijn er verschillende mogelijkheden, dan ontstaan een aantal alternatieve koncepten.
Per koncept wordt nu op basis van de verkeersattributen het netwerk ontworpen. Op basis van kosten, performance, uitbreidbaarheid en dat soort zaken wordt een van de koncepten geselekteerd en verder uitgewerkt om te komen tot de bouw van het netwerk.
In het kader van een informatieplan of automatiseringsplan gaat het om de keuze van de infrastruktuur. In grote lijnen gaat het dan om de lokale netwerken (LAN`s) en de netwerken om systemen in vestigingen met elkaar te verbinden (WAN`s).

Bij LAN`s gaat het om zaken als clustercontrollers, V24-kabels, PABX of coax-kabelnetwerken. Bij dit soort netwerken gaat het in de praktijk nooit om responsetijdproblemen of kapaciteitsproblemen. Transaktie analyse is hier alleen nodig om de kapaciteit te beheren. Verkeersattributen behoren standaard aanwezig te zijn om uitvoering via een WAN snel te kunnen beoordelen. De keuze van het LAN wordt meestal bepaald door fysieke zaken zoals de hoeveelheid kabels in de kabelgoten. Bij netwerkontwerp op basis van gegevensdistributie gaat het om WAN`s. Daarbij wordt een keus gedaan tussen een eigen netwerk of het openbare Datanet 1. In beide gevallen is Transaktie analyse nodig voor de bepaling van de hoeveelheid verkeer.
Op deze manier ontstaat een netwerk volgens dezelfde fasen van een projektaanpak als de toepassingen: analyse, logisch ontwerp, technisch ontwerp en bouw. Bovendien is dat netwerk gebaseerd op de informatiebehoeften van de gebruikers. Daarmee zijn de twee witte vlekken, genoemd in de inleiding, ingevuld.

37.8 Netwerkontwerp in distributieve omgevingen

In distributieve omgevingen gaat het om het verschil in topologie tussen gegevensgebruik en gegevensopslag. Er zijn n lokaties waar gegevens worden gebruikt en m lokaties waar gegevens kunnen worden opgeslagen. Daarbij kan n groter zijn dan m en omgekeerd, en ze kunnen geografisch geheel of gedeeltelijk samenvallen. Als voor een bepaalde gegevensverzameling geldt $m = 1$, dan is er verkeer nodig tussen lokaties om het gebruik mogelijk te maken. In het andere uiterste is $m = n$ en vallen ze geografisch samen. Dan is er verkeer nodig tussen de lokaties om de gegevensverzameling op alle lokaties konsistent te houden.
Naast dit soort technische problemen, spelen er nog zaken als eigendom van gegevens, beheer van gegevens, de bestaande situatie, de organisatie, de bedrijfspolitiek enzovoort. Kortom, als het al moeilijk is een gegevensmodel of een database-ontwerp af te stemmen op wensen van gebruikers, dan wordt het ontwerp van gedistribueerde databases een onoverkomelijk probleem. Dat betekent dat alle middelen, waarvan het nut en de noodzaak al jaren gepredikt wordt, in distributieve omgevingen onmisbaar zijn. Er moet een van het bedrijfsbeleid afgeleid informatieplan worden opgesteld. In het bedrijfsbeleid moeten in ieder geval de punten voorkomen die nodig zijn om een informatieplan voor gegevensdistributie op te stellen, zoals

- konkrete doelen die bereikt moeten worden met automatisering,
- de zelfstandigheid van de vestigingen, zonodig per aktiviteit of funktionele eenheid, uitgedrukt in verantwoordelijkheden en bevoegdheden ten aanzien van de gegevens,

- de plaats van het centrale rekencentrum ten aanzien van beheer en apparatuur keuze,
- de taken en bevoegdheden van ontwikkelingsafdelingen ten opzichte van gebruikers,
- de keuze mogelijkheden bij invoering van de automatisering.

In het informatieplan moet ten aanzien van de distributie van gegevens per groep gegevens geregeld worden wie de eigenaar/beheerder is en wie gebruikers zijn.
Tijdens het logisch ontwerp wordt door het transaktie-ontwerp vastgelegd wat iedere funktionaris via zijn beeldscherm met gegevens mag doen (Fig. 37.9). Waar de gegevens fysiek zijn opgeslagen is daarbij meestal niet van belang als het eigenaarschap maar goed is geregeld. Met behulp van decentrale transaktieschema's kan iedere eigenaar vastleggen wat er op andere lokaties met zijn gegevens gebeurt. Op een decentraal transaktieschema wordt de verwerking door de computer uitgebreid met het verkeer naar en de verwerking op een andere computer inclusief wat nodig is om databases konsistent te houden. Voor de gebruiker is dat meestal implementatie en dus ontstaan die schema's tijdens het technisch ontwerp. Ze kunnen tijdens het logisch ontwerp gemaakt worden door gebruiker en informatie-analist als de eerste het gebruik van gegevens op andere lokaties beschreven wil zien.
Tenslotte kan nog worden aangegeven wat de beperkingen of de mogelijkheden zijn van de topologie van de gegevensopslag, om het aantal alternatieven zoveel mogelijk te beperken. Daarbij zal de lokatie van reeds aanwezige systemen vaak een grote rol spelen.
Kortom, tijdens het logisch ontwerp wordt in het transaktie-ontwerp de topologie van het gegevensgebruik vastgelegd, zie Fig. 37.9: Funktionaris-transakties-werkplek.
Tijdens het technisch ontwerp wordt de topologie van het gegevensgebruik naast enkele mogelijkheden voor de topologie van de gegevensopslag gelegd. Per mogelijkheid ontstaat nu een set van centrale en decentrale transaktieschema's. In de literatuur (7,38) wordt gewerkt met factor tables om argumenten in kaart te brengen voor centralisatie en decentralisatie. In (7) worden matrices voorgesteld om de relatie tot stand te brengen tussen gegevens, processen en lokaties. Transktieschema's maken dat niet overbodig, maar leveren er een bijdrage in. Ook in distributieve omgevingen past transaktie-ontwerp tussen database-ontwerp en programma-ontwerp. Het staat vast dat het ontwerp van distributieve gegevensverwerking iets te maken heeft met netwerkontwerp, -belasting en -kosten. Toch wordt nergens konkreet aangegeven wat het verband ertussen is. In (7) worden matrices getekend waar per lokatie per proces wordt aangegeven welke databases worden gebruikt en met welke frequenties. Bij zo'n matrix staat dan de opmerking (pag. 399): The figures are traffic volumes in hundreds

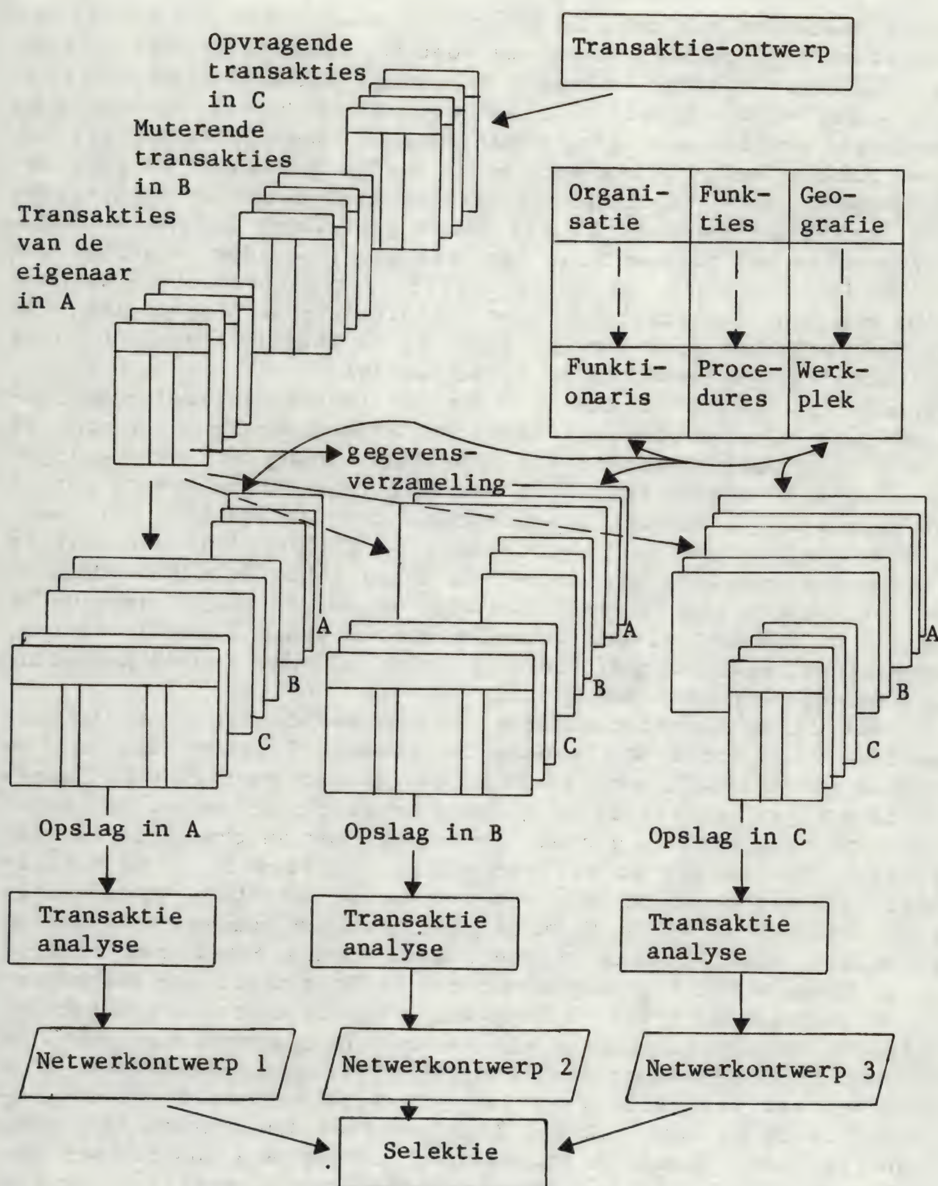


Fig. 37.10 Netwerkontwerp in een distributieve omgeving.

of transactions per day. Er wordt niet aangegeven hoe de cijfers tot stand zijn gekomen en wat ze voor het netwerk betekenen, hoewel de term "traffic volumes" wel enige verwachtingen wekt. In (7) wordt niets gedaan aan netwerkontwerp hoewel transmission costs als eerste overweging wordt genoemd (pag. 383). Dat kan ook niet anders, omdat alles gericht is op het gegevens- en programma-ontwerp. Voor bepaling van verkeer zijn denk- en wachttijden een onmisbaar gegeven. In (7) komen gebruikers alleen voor als uitvoerders van transactions (on databases) per day. Afgezien van de genoemde cijfers, is alles in (7) over distributieve verwerking een puur kwalitatieve, maar wel uitstekende beschouwing, die bijvoorbeeld een goed inzicht geeft in de mogelijkheden voor distributie van gegevens, en de konsekwenties.
Transaktieschema`s vervangen niets in systeemontwikkelingsmethoden, maar sluiten aan op andere ontwerpdokumenten en vormen de basis voor netwerkontwerp via Transaktie analyse. Per mogelijkheid voor de opslag worden de centrale transaktieschema`s geheel of gedeeltelijk omgezet in decentrale transaktieschema. Transaktie analyse levert per mogelijkheid de cijfers voor het verkeer tussen systemen. Op die manier ontstaan echte "traffic volumes" om een netwerk mee te ontwerpen. Per netwerkontwerp kunnen nu de aspekten in kaart worden gebracht die de keuze bepalen: kosten, apparatuur, performance, flexibiliteit, techniek van de koppelingen tussen computers enzovoort.
Als men in een vroeger stadium, bijvoorbeeld voor de projektselektie een inzicht wil hebben in de mogelijkheden dan kan in plaats van transaktie-ontwerp gekozen worden voor globale Transaktie analyse met als basis globale transaktieschema`s. Dat betekent dus toch wel een globaal onderzoek naar de topologie van het gebruik. Met minder is het onmogelijk. In Fig. 37.10 is het geheel in beeld gebracht voor een eenvoudige situatie. In de vestigingsplaatsen A, B en C wordt een bepaalde gegevensverzameling gebruikt. Deze gegevens zouden kunnen worden opgeslagen in A, B of C. Tijdens het transaktie-ontwerp is Transaktie analyse uitgevoerd voor iedere transaktie op basis van de centrale transaktieschema`s en is vastgesteld wie wat met de gegevens mag doen. Er is een eigenaar en er zijn andere gebruikers. Transakties zijn gebonden aan werkplekken en dus geografisch bepaald. Rechts bovenaan is de relatie met Fig 37.9 nog even aangegeven: een funktionaris voert bepaalde transakties uit op een geografisch bepaalde werkplek. Als de gegevens worden opgeslagen in A, moeten de centrale transaktieschema`s voor werkplekken in B en C worden omgezet in decentrale schema`s. Bij opslag in B geldt hetzelfde voor de schema`s van A en C enzovoort. Zo ontstaan uit de oorspronkelijke reeks transaktieschema`s drie reeksen transaktieschema`s. Per reeks wordt per decentrale transaktie, het detail-

schema aangepast. De nieuwe verkeersresultaten maken het mogelijk per reeks een netwerk te ontwerpen. De alternatieven worden tegen elkaar afgewogen en een ervan wordt gegebouwd.
In de praktijk hebben we vaak te maken met een rijdende trein: er zijn al systemen in gebruik, er bestaan al databases, er is al een netwerk. In principe maakt dit het ontwerpen eenvoudiger: er zijn minder vrijheidsgraden, de aanpak blijft hetzelfde. Wat het netwerk betreft zullen de resultaten van Transaktie analyse nu gebruikt worden om te beoordelen in hoeverre het bestaande netwerk gebruikt kan worden voor het distributieve verkeer. Dan moet wel de huidige belasting van de lijnen bekend zijn en zo niet dan moet die eerst worden vastgesteld, bijvoorbeeld op de manier die in het deel voor de transaktie-analisten wordt behandeld. (55.2)
Behalve voor het netwerkontwerp levert Transaktie analyse ook een bijdrage in het bepalen van de systeembelasting van de verschillende systemen. Ieder transport van gegevens door het netwerk betekent immers een stukje systeembelasting voor de computer aan de andere kant.
Samengevat kunnen we vaststellen dat het ontwerpen van gedistribueerde systemen een zeer ingewikkelde zaak is waarbij, in vergelijking met centrale toepassingen de eisen van de gebruikers moeilijker boven water te krijgen zijn omdat ze vaak subjektief zijn (het hebben van een eigen systeem, het baas in eigen huis zijn) en omdat het voor andere gebruikers weer volkomen onbelangrijk is waar de gegevens zijn opgeslagen, als de goedkoopste oplossing maar wordt gekozen. Met behulp van transaktie-ontwerp worden nu in ieder geval een paar zaken voor automatiseerders en gebruikers duidelijk: welke funktionarissen bepaalde transakties mogen uitvoeren, daarmee is het beheer en het eigendomsrecht voor gebruikers goed duidelijk en ten aanzien van de mogelijkheden voor het netwerkontwerp kunnen nu een aantal situaties konkreet worden doorgerekend. Ook hier werkt de terugkoppeling met cijfers verhelderend. Als gebruikers om allerlei redenen niet tot uitspraken komen over de gegevensopslag wordt op grond van de transaktieschema's een aantal schattingen gedaan en het rekenproces uitgevoerd. De cijfers werken dan vaak als een katalysator voor beslissingen en keuzes.

DEEL 4
voor informatie-analisten

Prototyping is door automatiseerders uitgevonden, dialoogsimulatie had door gebruikers bedacht kunnen worden.

Hoofdstuk 41
Mensen, methoden, middelen

41.1 Taakomschrijving en vakmanschap

De funktie "informatie-analist" is vastgesteld door het Nederlands Genootschap voor Informatica. In het kader van het ontwerpen van interaktieve toepassingen en het toepassen van bepaalde methoden zullen we de taak wat nauwkeuriger omschrijven. Dat betekent niet dat daarmee iets wordt afgedaan aan bestaande omschrijvingen. Het gaat om detaillering en toevoegingen. Het aksent ligt daarbij op de kommunikatie met de gebruiker. Tijdens de analysefase brengt de informatie-analist de bestaande situatie in kaart. Tijdens het logisch ontwerp is hij weer bij de gebruiker, maar nu om samen met hem de transakties te ontwerpen. In sommige bedrijven worden informatie-analisten gerecruteerd uit gebruikersafdelingen. In dat geval is er een uitgebreide materiekennis aanwezig, wat een vlotte kommunikatie met gebruikers mogelijk maakt. In andere situaties zijn informatie-analisten voortgekomen uit de eigen automatiseringsafdeling of ze zijn afkomstig van externe software houses. Dan is er een volledige scheiding tussen materiekennis en automatiseringskennis. Van die situatie gaan we uit. De informatie-analist moet beschikken over praktische kennis op een aantal terreinen, ervaring hebben met methoden en gereedschappen. In het kader van de te behandelen methoden betekent dat het volgende:

- Hij moet beschikken over voldoende parate kennis van ergonomie

en dialoogontwerp. Er is een groot aantal boeken geschreven over deze onderwerpen, maar gezien de praktijk hebben ze niet veel effekt. Een goede informatie-analist moet alle bekende dialoogvormen met hun voor- en nadelen uit z'n hoofd kennen, evenals de basisregels voor schermontwerp. Dat is de enige mogelijkheid om het ter plekke alternatieven te bedenken waaruit de gebruiker kan kiezen. Juist een methode als dialoogsimulatie biedt de mogelijkheid om snel alternatieven te demonstreren en te evalueren.

- Hij moet de funkties van het transaktieschema goed kennen en er rekening mee houden. Het transaktieschema is een gebruikersdokument: de informatie-analist moet de gebruiker zover krijgen dat die het transaktieschema maakt. Lukt dat niet, dan moet hij ervoor zorgen dat de taal op het schema zodanig is dat de gebruiker er volledig achter kan staan. Het transaktieschema moet kunnen dienen om ontworpen gegevensmodellen en funktiemodellen aan te passen. Niet door bijvoorbeeld een algoritme volledig te beschrijven, maar door precies aan te geven welk algoritme bedoeld wordt. Hetzelfde geldt voor de gegevens.

Het transaktieschema is het startdokument voor Transaktie analyse. De informatie-analist moet weten hoe transaktieschema's verwerkt worden door de transaktie-analist. In veel gevallen wordt het transaktieschema gebruikt als interface-dokument tussen informatie-analist en transaktie-analist. Soms gaat het daarbij om de overgang van logisch naar technisch ontwerp.

Tenslotte, en dat is voor de informatie-analist het belangrijkste aspekt, is het transaktieschema de start voor de dialoogsimulatie. Steeds weer proberen informatie-analisten dialoogsimulatie uit te voeren zonder transaktieschema. Aan het eind van het logisch ontwerp is er dan een pak schermlay-outs beschikbaar, maar de gebruiker is de relatie met de handmatige aktiviteiten kwijt en er is geen basis om Transaktie analyse uit te voeren of de gegevens- en funktiemodellen aan te passen. Een goede informatie-analist kent alle aspekten van het transaktieschema en zorgt ervoor dat de gebruiker ook goed weet waar het omgaat.

- Hij kent dialoogsimulatie als methode.

Hij weet het verschil met prototyping en kent de beperkingen van zijn simulator. Hij weet dat het doel van dialoogsimulatie de volledige participatie is van de gebruikers in het automatiseringsgebeuren. Hij zorgt ervoor dat dat doel bereikt wordt ondanks allerlei vooroordelen en ongeinteresseerdheid van de gebruikers.

- Hij overziet de iteratieve aspekten van het ontwerpproces en brengt de gebruiker van vrijblijvende probeersels naar definitieve eisen voor het ontwerp.

- Hij begrijpt waarom het ontwerpen van interaktieve toepassingen niet begint met het maken van schermlay-outs. Hij trekt zich

niets aan van collega's, die nog vastzitten aan de oude manier van batch-toepassingen ontwerpen. Die zijn nooit verder gekomen dan het vervangen van de printlay-out door de schermlay-out, waarbij de inspraak van de gebruiker hetzelfde is gebleven.
- Ondanks de komplexiteit van vele systemen blijft hij de rode draad in de gaten houden. Hij laat zich niet van de wijs brengen door allerlei details van de gebruikers noch door de planningsdruk van de automatisering. Temidden van een groot aantal komplexe transakties weet hij de 20-80-regel korrekt toe te passen, ziet hij het verschil tussen zware en lichte transakties en stelt hij vast wat geschat kan worden en wat nader onderzocht dient te worden.
- Hij heeft voldoende flexibiliteit om zich aan te passen aan de konkrete situatie waarin een projekt verkeert. Wanneer er om allerlei redenen bijvoorbeeld onvoldoende gelegenheid is met alle gebruikers te kommuniceren, blokkeert hij de voortgang niet, maar rapporteert hij direkt de konsekwenties. Dan mag het projektmanagement kiezen voor de voortgang van het ontwerpproces of de konsekwenties achteraf.
- Hij ziet achter ieder dialoogontwerp een paar alternatieven. Wanneer het transaktieschema is gemaakt kan hij de haalbaarheid van het ontwerp inschatten. Hij weet zoveel van interaktieve toepassingen af dat hij aanvoelt of iets 2 seconden of 20 seconden zal duren. Wanneer hij voorziet dat een transaktie nooit aan de responsetijdverwachtingen van de gebruiker kan voldoen, zoekt hij naar alternatieven. Daarbij gaat hij zonodig van de ontwerpfase terug naar de analysesfeer.
- Hij motiveert de gebruiker om kreatief om te gaan met de mogelijkheden die de computer biedt. Op die manier legt hij de basis voor een stuk funktieverruiming in de toekomst. Uiteraard houdt hij daarbij de definitie van het huidige projekt in de gaten.
- Hij kan omgaan met gebruikers op diverse niveau's. Hij presenteert op korrekte wijze de samenwerkingsvorm die nodig is voor het ontwerpen van interaktieve toepassingen.
- Wanneer er iets schort aan de samenwerking met de gebruiker, voelt hij aan of het gaat om onmacht of onwil. Wanneer een gebruiker bepaalde cijfers niet heeft, kan er gezocht worden naar mogelijkheden om ze te vinden. Wanneer hij ze niet wil geven, duidt dat op een probleem van een heel andere orde. De oorzaken van die problemen kunnen van bedrijfspolitieke of persoonlijke aard zijn. In het algemeen is het niet verstandig om te proberen bedrijfspolitiek aan te pakken via automatisering. Bij persoonlijke problemen is het zaak om niet klakkeloos door te gaan met het projekt. Het is mogelijk dat de problemen kunnen worden opgelost door de projektaanpak en de inbreng van gebruikers goed naar voren te brengen, maar het is ook mogelijk dat de problemen die-

per liggen en dat personeelszaken moet worden ingeschakeld. Dat moet dan wel tijdens het logisch ontwerp gebeuren! De informatie-analist geeft dus tijdig een signaal en doet wat er binnen het kader van zijn taakstelling mogelijk is.

- Hij weet helder en zakelijk te rapporteren. In de automatisering zijn overal dokumenten voor, behalve voor de kommunikatie met de gebruikers. Het initiatief tot rapportage ligt daardoor bij de informatie-analist. Er zijn ook bedrijven waar de hoeveelheid rapporten niet te overzien is. De informatie-analist zoekt de gulden middenweg, houdt per transaktie de zaak in de gaten en rapporteert de onduidelijkheden en stoornissen in de kommunikatie met de gebruikers en eventuele konsekwenties zowel in technisch als in ergonomisch opzicht.

Kortom, een zware funktie met veel aspekten. Hoezeer we ook zoeken naar automatisering van de automatisering, deze funktie zal altijd door deskundigen moeten worden vervuld. Daarnaast moet worden vastgesteld dat niet iedere programmeur per definitie naar deze funktie kan doorgroeien. Materiedeskundigen met enige jaren ervaring in de systeemanalyse en het ontwerpen van interaktieve toepassingen zouden wel eens de beste informatie-analisten kunnen zijn. In een zin samengevat: hij vult de bovenste witte vlek in, die is aangegeven in de proloog.

41.2 Methoden en middelen

Een methode is een uitgekristalliseerde, gestandaardiseerde en goed gedokumenteerde manier van werken. Bij een methode kunnen middelen, gereedschappen horen. Het interviewen van gebruikers is in deze zin dus geen methode. De manier van interviewen is niet gestandaardiseerd, de wijze van rapportage is vrij. Hoewel standaardisatie een negatieve klank kan hebben en als een harnas kan aanvoelen, is ze toch noodzakelijk in situaties waarin veel mensen met elkaars produkten moeten werken. In de automatisering is dat alleen al tengevolge van de specialisatie het geval.

Vaak is het bij een methode zo, dat diverse aktiviteiten elkaar opvolgen. Zo kunnen diverse middelen worden toegepast die leiden tot konklusies, die weer kunnen leiden tot iteraties. We zullen dit voor het ontwerpen van transakties in deze paragraaf duidelijk maken.

Het invoeren van methoden in een organisatie is een probleem op zich. Iedereen kent eigenlijk maar een goede methode en dat is zijn eigen manier van werken. Omschakeling is een veranderingsproces dat iedereen altijd veel moeite kost. De methoden waar het nu omgaat betekenen slechts voor een deel veranderingen, het gaat meer om toevoegingen aan bestaande methoden. In Fig 41.1 is dat aangegeven. Het gaat in deze figuur alleen om de grote lijnen.

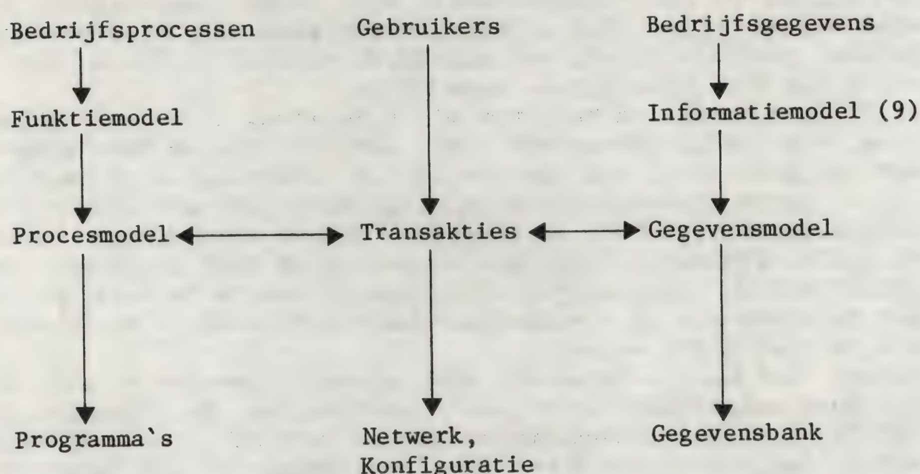


Fig. 41.1 De plaats van methoden voor transaktie-ontwerp

Het is duidelijk dat de gebruikers ook betrokken zijn bij de analyses, zoals die altijd plaatsvinden. Deze figuur dient alleen om de toevoeging aan bestaande methoden aan te geven.
De gebruiker ontwerpt samen met de informatie-analist transakties. Via de standaarddokumenten worden transakties en modellen met elkaar in overeenstemming gebracht. Daarna verlopen programma- en gegevensontwerp op de normale manier. Daarnaast vormen de transakties de basis voor het netwerk en de konfiguratie. Een en ander betekent dat de te beschrijven methoden in elke projektaanpak kunnen worden opgenomen. We zullen nu de methoden aangeven met hun aktiviteiten en het gebruik van de middelen, in dit verband gereedschappen, tools. Hoe de aktiviteiten worden verdeeld over de verschillende fasen wordt behandeld in het hoofdstuk Transakties. Transaktie analyse kan op drie manieren worden uitgevoerd:
- Ergonomische Transaktie analyse. Bij deze analyse ligt het aksent op de procedure aan het beeldscherm, de verwerkingsaspekten worden nauwelijks of niet in het detailschema tot uitdrukking gebracht. De resultaten leveren cijfermateriaal op voor de informatie-analist.
- Logische Transaktie analyse. Bij deze analyse worden de verwerkingsaspekten in het detailschema opgenomen voor zover die bekend kunnen zijn tijdens het logisch ontwerp. De resultaten zijn van

belang wanneer het gaat om duidelijke afwijkingen van de verwachtingen.
- Technische Transaktie analyse. Nu wordt het detailschema gedetailleerd en volledig ingevuld. De resultaten zijn de basis voor netwerk- en konfiguratie ontwerp.
De ergonomische en de logische analyses kunnen ook in globale vorm worden uitgevoerd, op basis van globale transaktieschema's. Natuurlijk zijn de resultaten dan ook globaal, maar ze kunnen al tot interessante konklusies leiden.
Transaktie analyse wordt uitgevoerd door transaktie-analisten, de resultaten van de logische en technische Transakties analyse worden eveneens door hen verwerkt. De methode voor transaktie-ontwerp valt in twee op elkaar aansluitende methoden uiteen: Dialoogsimulatie en Transaktie analyse.

- Methode: Dialoogsimulatie
  - Aktiviteit: Ontwerpen van transakties
  - Middel : Transaktieschema
  - Aktiviteit: Simuleren van de transakties
  - Middel : Dialoogsimulator
  - Aktiviteit: Evalueren van resultaten
  - Aktiviteit: Resultaten van dialoogsimulatie vergelijken met modellen, standaards, etc.
  - Aktiviteit: Resultaten van dialoogsimulatie invoeren in het ontwerpproces
- Methode: Ergonomische Transaktie analyse
  - Aktiviteit: Opstellen van het detailschema op basis van het transaktieschema en uitvoeren van het rekenproces
  - Middelen : Detailschema, rekenprogramma
  - Aktiviteit: Resultaten omrekenen naar konsekwenties voor gebruikers
- Methode: Logische Transaktie analyse
  - Aktiviteit: Aanpassen of maken van detailschema's en uitvoeren van het rekenproces
  - Middelen : Detailschema, rekenprogramma
  - Aktiviteit: Resultaten omrekenen naar konklusies voor gebruikers en automatiseerders
- Methode: Technische Transaktie analyse
  - Aktiviteit: Aanpassen van de detailschema's en uitvoeren van het rekenproces
  - Middelen : Detailschema, rekenprogramma
  - Aktiviteit: Resultaten omrekenen naar konsekwenties voor konfiguratie en netwerkontwerp

Als de verschillende vormen van Transaktie analyse na elkaar worden uitgevoerd, gaat het steeds om hetzelfde detailschema, dat wordt aangepast en om hetzelfde rekenprogramma. Uit deze opsomming blijkt dat op diverse momenten het iteratieve aspekt in het

ontwerp geaksentueerd wordt. Overal waar resultaten verschijnen kan ofwel binnen de automatisering ofwel via de gebruikers een iteratie gestart worden. Bij de behandeling van beide methoden werken we dit iteratieve aspekt verder uit.
Wanneer de details van de methoden na de lezing van de volgende hoofdstukken duidelijk zijn, zal blijken dat sommige onderdelen vastliggen door de manier van werken, maar dat het belangrijkste, het ontwerpen, een zaak van de informatie-analist zelf blijft. Zijn vakmanschap behoort niet te blijken uit een zelf ontworpen formulier voor het transaktieschema, maar uit een transaktie-ontwerp dat ergonomisch verantwoord is en waar de gebruikers enthousiast over zijn. Het is verbazingwekkend te zien hoe sommige informantie-analisten zich kunnen verliezen in papieren details en tegelijkertijd ontwerpen afleveren die qua ergonomie nog stammen uit het batch-tijdperk.

41.3 Relaties tussen methoden

In Fig. 41.1 zijn de relaties aangegeven tussen transaktie-ontwerp en gegevens- en funktie-ontwerp. Dat zijn relaties in horizontale zin. In deze paragraaf gaat het om de relaties tussen methoden binnen transaktie- en netwerkontwerp, zoals aangegeven in Fig. 41.2. Onafhankelijk van de simulatiemethode die er op volgt, begint het transaktie-ontwerp altijd met het transaktieschema. Het transaktieschema geeft transakties weer zoals de gebruiker ze herkent en in een taal die hij begrijpt. Bij die herkenning gaat het om twee aspekten: de aansluiting op handmatige procedures en het kunnen aangeven van kwantiteiten. Voor de gebruiker is het transaktieschema het belangrijkste dokument en vaak ook het enige dat in zijn taal geschreven is. De procedure die beschreven is op het transaktieschema moet de gebruiker herkennen zo dat hij kan zeggen hoe vaak hij die per dag uitvoert, waarvan dat aantal afhankelijk is en wat de pieken zijn.
Als er verder niets gesimuleerd zou worden, dan is het transaktieschema het dokument dat wordt overgedragen aan de transaktie-analist. Hij maakt op basis daarvan een detailschema. In feite is dat het gekwantificeerde transaktieschema, geschikt voor invoer in het rekenprogramma. Wanneer geen dialoogsimulatie wordt gedaan zijn er voor het maken van het detailschema twee mogelijkheden wat de velden op het scherm betreft: de aantallen tekens worden geschat of het detailschema wordt pas gemaakt wanneer de schermlay-outs bekend zijn. Wanneer het gaat om een ergonomische Transaktie analyse, is het aantal te displayen tekens niet van belang. Te allen tijde geldt voor het rekenprogramma: garbage in, garbage out. De nauwkeurigheid van de resultaten hangt alleen af van de nauwkeurigheid van het detailschema.

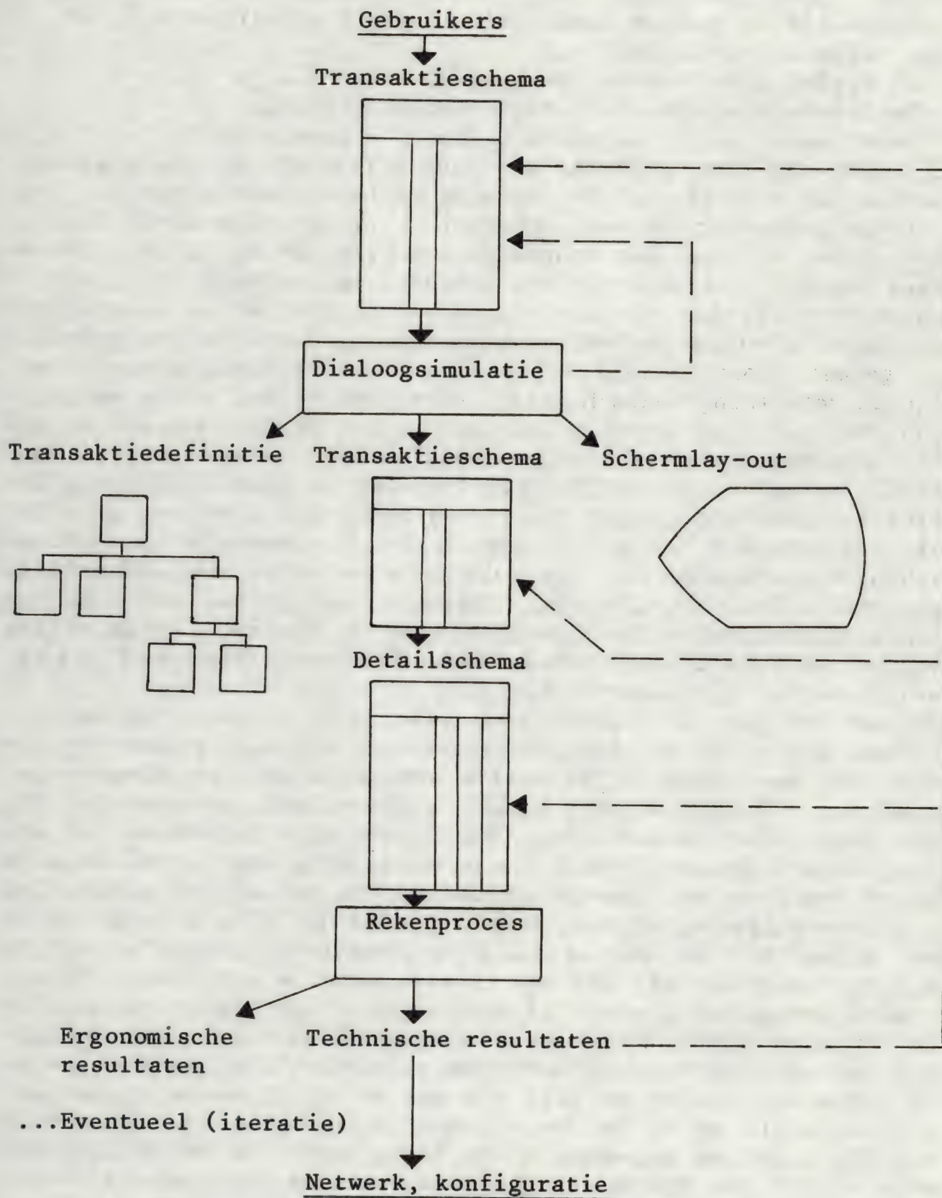


Fig. 41.2 Vertikale relaties binnen transaktie-ontwerp

Als het goed gaat vormt het transaktieschema de basis voor dialoogsimulatie. Wanneer dialoogsimulatie is gerealiseerd is er het volgende beschikbaar
- de definitieve transaktieschema`s,
- beeldschermlay-outs inclusief velddefinities,
- de volgorde van de schermen tijdens de simulatie.

Op basis van deze gegevens kan zeer effektief een begin gemaakt worden met prototyping. Prototyping is immers ook gebaat bij duidelijke gebruikersinbreng? Starten we prototyping op deze wijze dan is het voordeel dat Transaktie analyse, en dus de kwantificering van het hele interaktieve gebeuren, gehandhaafd blijft. Prototyping heeft het voordeel dat de verwerking en de toegang tot de gegevens vroeg in het ontwerpproces in kaart worden gebracht en getest. Voor Transaktie analyse betekent prototyping alleen dat de verwerking waarschijnlijk nauwkeuriger kan worden aangegeven dan zonder prototyping. Evaluatie van prototyping kan leiden tot aanpassing van transaktieschema`s. Dat betekent dat de volgende versie van de definitieve transaktieschema`s ontstaat. Definitief betekent (eigenlijk alleen maar) dat ze worden gebruikt of gebruikt zijn om detailschema`s mee te maken. De transaktieschema`s waarmee de dialoogsimulatie begon waren niet definitief. Het maken van detailschema`s gebeurt altijd aan de hand van transaktieschema`s. De uitzonderingen op die regel zijn alleen van belang voor transaktie-analisten en komen alleen voor bij de evaluatie van bestaande systemen.

Wanneer het gaat om ergonomische resultaten, wordt in het detailschema alleen de bedieningsaspekten van het beeldscherm opgenomen. Bij een logische Transaktie analyse worden de verwerkingsaspekten op basis van een logisch gegevensontwerp opgenomen. Bij een technische Transaktie-analyse worden alle parameters van een transaktie opgenomen zodat ook de technische resultaten verwerkt kunnen worden. Het rekenprogramma levert in alle drie gevallen dezelfde gegevens, alleen zijn die niet in alle drie situaties van belang voor de konklusies. De ergonomische Transaktie analyse levert resultaten die van belang zijn voor de gebruikers. En logische Transaktie analyse is gebaseerd op een logisch ontwerp en levert de verwachte technische resultaten op. Vallen die ver buiten het verwachtingspatroon van de gebruiker, dan kan daar meteen op gereageerd worden en niet pas aan het eind van de implementatiefase zoals nu in de meeste projekten. Technische Transaktie analyse levert de gegevens op die nodig zijn voor het netwerkontwerp en ook dan levert het programma weer de ergonomische resultaten op. Misschien wijken die inmiddels af van eerdere resultaten en dan is kommunikatie met de gebruiker nog altijd op z`n plaats. Liever tijdens het technische ontwerp dan tijdens de implementatie.

Hiermee zijn de vertikale lijnen aangegeven. In de volgende paragraaf worden de horizontale lijnen beschreven. Beide worden in volgende hoofdstukken verder uitgewerkt.

41.4 Projektaanpak

De meeste automatiseringsprojekten verlopen op de bekende manier: analyse, ontwerp, bouw, invoering, produktie, evaluatie. In veel bedrijven is gekozen voor een van de vele systeemontwikkelingsmethoden. Hoewel deze methoden in grote lijnen allemaal op hetzelfde neerkomen zijn de onderlinge verschillen aanzienlijk. De ene methode omvat bijna alleen aktiviteiten die in de analysefase van een projekt plaatsvinden. Bij een andere methode vormt het logisch ontwerp 90% van het geheel. Bij sommige methoden ligt het aksent op de dokumentatie. Er is geen methode die alle aspekten van alle fasen volledig in zich heeft. (9 pag. 528) Bovendien is het zo, dat veel bedrijven die kiezen voor een methode, die toch aanpassen aan eigen methoden of dokumentatiesystemen.
Kortom, zoals overal in de automatisering, vinden we ook hier weinig algemeen toegepaste standaards.
Dat betekent dat we bij invoering van de methode transaktie-ontwerp, per bedrijf moeten bekijken hoe de aansluiting met bestaande methoden moet plaatsvinden. In het hoofdstuk Transakties gaan we in op de details, nu gaat het om de grote lijnen. We beginnen met de transaktieschema's. We gaan er in dit hoofdstuk van uit dat de transaktieschema's gemaakt worden op het meest voor de hand liggende moment: tijdens het logisch ontwerp. In de meeste gevallen is er kontakt geweest met de diverse gebruikersgroepen tijdens de analysefase. Zorg ervoor dat de gebruikers goed duidelijk is wat het verschil is tussen analyse en logisch ontwerp. Sluit aan bij de informatie die de gebruikers verstrekt hebben tijdens de interviews. Bepaal op welke manier de resultaten van analysefase worden vastgelegd. In het algemeen worden processen en gegevens beschreven, soms met kwantiteiten, pieken, geografische aanduidingen en dergelijke, maar voor de gebruiker moet het transaktieschema een logisch vervolg zijn op de analysefase. Voor de projektaanpak moet de informatie-analist op de hoogte zijn van de analyse resultaten, maar het transaktieschema is een startdokument dat pas later gekoppeld wordt aan andere projektaktiviteiten -dokumenten. Wanneer er al transaktieschema-achtige dokumenten voorhanden zijn, dan kunnen die gebruikt worden wanneer ze aan de volgende eisen voldoen:
- leesbaar voor de gebruikers, alsof ze door hen waren opgesteld.
- volledige weergave van de aansluiting tussen handmatige aktiviteiten en het werken met het beeldscherm: aan- en uitloop, denk- en wachttijden.

- geschikt om een detailschema mee te maken.
- geschikt om dialoogsimulatie op te baseren.

In de meeste gevallen blijkt echter dat bestaande ontwerpdokumenten automatiseringsdokumenten zijn, die helemaal niet leesbaar zijn voor gebruikers. Voorzover het de gebruikers betreft, kan de funktie van het transaktieschema in een woord worden samengevat: gebruikersontwerpdokument. Voor veel automatiseerders een contradictio in terminis! In de meeste systeemontwikkelingsmethoden komt zo'n dokument dan ook niet voor. Gebruikers ontwerpen immers niet, laat staan dat ze dokumenten ervan in hun bezit hebben. In de meeste gevallen moeten transaktieschema's gemaakt worden als warme start van het ontwerp van interaktieve toepassingen.
Het transaktieschema vormt de basis voor dialoogsimulatie. In sommige bedrijven is het automatiseren van de automatisering zover gevorderd, dat er zoveel mogelijk informatieverwerkende systemen bij worden ingezet. Dat betekent dat de dialoogsimulator moet worden ingepast in het geheel van bestaande gereedschappen. Daardoor wordt bijvoorbeeld bepaald of een dialoogsimulator een vrijstaande micro mag zijn of dat het een pakket is op een ontwikkelcomputer. In het hoofdstuk Dialoogsimulatie gaan we daar verder op in. Als we letten op de dokumenten of resultaten van dialoogsimulatie, wordt duidelijk dat de koppeling met bestaande methoden niet ingewikkeld kan zijn.
De schermlay-outs worden altijd gemaakt. Meestal op het verkeerde moment en onbegrijpelijk voor de gebruiker, maar iedere ontwerper weet ze te plaatsen in de projektaanpak. De schermlay-outs die afkomstig zijn van dialoogsimulatie kunnen dus op de standaardmanier worden verwerkt. Dat hoeft overigens niet te betekenen dat na dialoogsimulatie alle schermlay-outs beschikbaar zijn. Dialoogsimulatie is er primair voor de gebruikers. De informatieanalist kan dus best besloten hebben om van een aantal transakties die bijna hetzelfde zijn, er met de gebruiker enkele te simuleren. Hoe de ontbrekende schermlay-outs gemaakt worden is in dit verband niet van belang. De informatie-analist moet zich wel realiseren dat er geen Transaktie analyse wordt uitgevoerd voor transakties waarvan geen transaktieschema aanwezig is. Een volgend produkt van dialoogsimulatie is een overzicht per schermlayout van gebruikte velden, hun funktie, hun lengte, hun displayattributen enzovoort. Er zijn twee uitersten denkbaar. Het ene uiterste is de situatie waarin alle gegevens over gegevens vastliggen en vastgelegd zijn in dokumenten of data dictionaries. In dit geval moet de informatie-analist die informatie gebruiken bij het maken van het startontwerp op de simulator. Het andere uiterste is de situatie waarin de informatie-analist de dialoogsimulator naast het eigenlijke doel, ook gebruikt om gegevens over gegevens van de gebruikers los te krijgen en vast te leggen. Hoe de

situatie ook is, er zal voor gezorgd moeten worden dat de velden op beeldschermen passen op de velden in de records. Zelfs binnen de meest primitieve projektaanpak zijn recordindelingen te vinden. Deze aansluiting kan dus nooit een probleem vormen.
Een volgend produkt van dialoogsimulatie is de transaktiedefinitie. Een transaktiedefinitie is de opeenvolging van schermen binnen een transaktie zoals die is gespecificeerd op de dialoogsimulator. Om de uiteindelijke interaktieve programma's te kunnen ontwerpen moet exakt bekend zijn hoe de schermvolgorde zal zijn, kompleet met alle sprongmogelijkheden. Het is nog beter als de hele dialoog formeel beschreven is in de vorm van een taal of een schematechniek. Dan zijn de schermlay-outs elementen binnen de beschrijving. Bezwaar van al dit soort technieken is dat ze voor de meeste gebruikers onleesbaar zijn. Als er transaktieschema's gemaakt zijn, vormen die uiteraard een prima basis onder allerlei vormen van dialoogbeschrijvingen. In (14) worden enkele methoden genoemd. Er moet in ieder geval een dialoogstruktuur ontworpen zijn die alle schermlay-outs bevat met hun onderlinge relaties. In veel systeemontwikkelingsmethoden ontbreekt een ontwerp- of tekentechniek. Hoe het ook zij, de transaktiedefinities van de dialoogsimulator vormen de basis voor deze strukturen. Alle transaktiedefinities tesamen kunnen echter in twee opzichten inkompleet zijn. We hebben al opgemerkt dat de informatie-analist, in overleg met de gebruiker, van een aantal bijna identieke transaktie er maar een paar simuleerde. Verder kan het zijn dat binnen transakties niet alle schermlay-outs zijn vastgelgd op de simulator. Zo zou bijvoorbeeld best de tekst op helpschermen op een heel ander moment in de projektaanpak bepaald kunnen worden. Het zou natuurlijk niet slecht zijn om te bepalen, dat alle transakties en alle schermlay-outs op de simulator ontworpen moeten worden, onafhankelijk van het feit of ze met de gebruiker gesimuleerd worden of niet. Het nut van zo'n afspraak hangt onder andere af van de plaats van de simulator in het hele ontwerpproces. Wanneer de koppeling tussen simulator en schermbibliotheken van progammeurs 100% is, is het een verstandige afspraak. Wanneer de simulator een "stand alone" gereedschap is en de schermlay-outs bijvoorbeeld geprint worden om vervolgens handmatig te worden ingevoerd in de schermbibliotheek van de ontwikkelcomputer, is het al minder duidelijk of het een verstandige afspraak is.
Vaak bestaan binnen een bedrijf een aantal standaards, soms mede tot stand gekomen door de invoering van een systeemontwikkelingsmethode. In het kader van het onderwerp bepalen we ons tot standaards ten aanzien van dialoogontwerp en schermlay-out. De standaards liggen per definitie vast en de informatie-analist moet ze kennen. Standaardisatie van schermlay-outs betekent meestal dat er een vaste kopindeling is en dat onderaan het scherm een aantal

regels zijn gereserveerd voor boodschappen of foutmeldingen. Menuschermen komen zo vaak voor, dat ze meestal ook gestandaardiseerd zijn. Verder zijn er meestal enige richtlijnen bekend ten aanzien van zaken als inverse video, blinking, full en half density. De informatie-analist kan bij het maken van zijn startontwerpen uitgaan van deze standaards. Als het goed is zullen de gebruikers er weinig problemen mee hebben. Is dat niet het geval, dan moeten de standaards nog eens bekeken worden. Ten aanzien van de standaards voor dialoogontwerp ligt de zaak wat anders. Een goede simulator moet de gebruiker een transaktie laten ervaren. Dat betekent soms dat de informatie-analist enige funktietoetsen van de simulator heeft gebruikt om iets te demonstreren wat in werkelijkheid door het programma gedaan zal worden. Met andere woorden, er is een intelligente vertaalslag nodig van de transaktiedefinitie naar een dialoogstruktuurdiagram dat aan de standaards voldoet.
Hoe groot die slag ook is, als de dialoogsimulatie funktioneel juist is uitgevoerd, moet die slag voor de gebruikers bijna onmerkbaar zijn. Het transaktieschema is immers de geaksepteerde basis onder de dialoogsimulatie en het uiteindelijke systeemontwerp.
Het definitieve transaktieschema is het transaktieschema dat na de dialoogsimulatie het transaktie-ontwerp korrekt weergeeft. Afgezien van de menselijke handelingen en de dialoog, is ook de verwerking door de computer omschreven. In gebruikerstermen is aangegeven welke funktie het programma moet uitvoeren en welke gegevens daar bij gebruikt moeten worden. Zoals in Fig. 41.1 is aangegeven loopt transaktie-ontwerp parallel met andere ontwerpaktiviteiten, die resulteren in gegevensontwerp en programma-ontwerp. De funkties die genoemd worden op het transaktieschema moeten terug te vinden zijn in het programma-ontwerp. Zo niet, dan moet het programma-ontwerp worden aangepast of het transaktieschema. Dat laatste kan alleen in overleg met de gebruiker die een kopie heeft van het transaktieschema. Hetzelfde geldt voor de gegevens. Het moet in principe mogelijk zijn dat een kreatieve gebruiker een nieuw gegeven bedenkt. Nieuw betekent in dit verband, dat het bij de analyse niet boven water is gekomen en niet is opgenomen in de gegevensstrukturen. Wanneer deze afstemming heeft plaatsgevonden is het transaktieschema afgehandeld wat de koppeling met andere projektaanpakaktiviteiten betreft. Voor de gebruiker wordt het transaktieschema weer effektief bij de oplevering. Dan beoordeelt hij de beeldschermsituatie aan de hand van de transaktieschema's. Dan worden de automatiseerders afgerekend op hun implementatie van gebruikersinbreng.
Het transaktieschema is tenslotte startdokument voor Transaktie analyse. Deze methode komt behalve in PARAET in geen enkele sys-

teemontwikkelingsmethode voor en dus is koppelen niet van toepassing. Samengevat kunnen we vaststellen dat transaktie-ontwerp parallel loopt met bestaande aktiviteiten en dat er in elke projektaanpak enkele zeer konkrete synchronisatiepunten te vinden zijn.

41.5 Methoden en omgeving

De omgeving waarin informatie-analist en transaktie-analist werken kan per projekt en per bedrijf sterk verschillen. In het ene bedrijf staat al jarenlang een groot mainframe voor batch-verwerking, in een ander bedrijf is men een paar jaar geleden al overgeschakeld op interaktieve toepassingen. Het ene bedrijf bestaat uit een vestiging, het andere is verspreid over vele plaatsen. In het ene bedrijf begint men net aan de automatisering met een minicomputer, in het andere bedrijf krijgt iedere medewerker een microcomputer. Het ene bedrijf beschikt over een staf van ervaren en deskundige automatiseringsspecialisten, terwijl het andere bedrijf net genoeg mensen in dienst heeft om een minicomputer aan de praat te houden. Hoewel een methode als Transaktie analyse altijd dezelfde soort cijfers oplevert zal de manier van werken van zowel de informatie-analist als de transaktie-analist per omgeving duidelijk verschillen. In een omgeving waar een netwerk moet worden ontworpen zullen de aksenten binnen Transaktie analyse anders liggen dan in een omgeving met een minicomputer en een paar lokale beeldschermen.
Om de verschillen in aanpak duidelijk te maken zullen we de omgevingen indelen. Er zijn natuurlijk vele manieren om bedrijven te rangschikken. De gekozen invalshoeken maken het mogelijk de verschillen in werkwijze en gebruik van de te beschrijven methoden te illustreren. We zullen de omgevingen indelen op basis van:
- Gebruik van methoden en middelen in het ontwerpproces. Dit is van belang voor de koppeling tussen diverse methoden. De gradatie is vrij grof, maar voldoende voor het doel. Buiten beschouwing blijft of de ontwikkeling van informatieverwerkende systemen door eigen mensen wordt gedaan of door ingehuurd personeel. Als de hele bemanning voor een projekt wordt ingehuurd of een projekt wordt uitbesteed, moet men vaststellen in welke mate men gebruik gaat maken van methoden en middelen. Een ingehuurde enkeling past zich natuurlijk aan aan de bestaande situatie.
- De computerverspreiding. Bij deze invalshoek gaat het om het aantal lokaties met een computer, het soort computer en de aanwezigheid van een netwerk. Het woord rekencentrum slaat op een computerruimte met een of meer mainframes en eventueel minicomputers, met daaromheen de hele bemanning voor operations, klantensupport, onderhoud, systeemprogrammering enzovoort. Bij een lo-

katie met een minicomputer is de bemanning en de organisatie er omheen meestal ook mini.

- Het soort projekt. Het bepalen van de belasting van een interaktieve toepassing met tien beeldschermen op een mini is iets anders dan de bepaling van de topologie van een netwerk, hoewel in beide gevallen Transaktie analyse wordt gebruikt. De indeling in projekten is niet uitputtend, maar wanneer voldoende ervaring is opgedaan met de methoden kan de informatie-analist of de transaktie-analist zelf nieuwe toepassingen beoordelen.
- Methoden en middelen
  - M1: Er wordt niet een voor iedereen geldende, vaststaande methode in de analysefase en het logisch en technisch ontwerp toegepast. Iedere informatie-analist en systeemontwerper kiest de manier van werken, die naar zijn idee het beste past bij het soort projekt.
  - M2: Er is gekozen voor een bepaalde notatiewijze in de analysefase. Er worden handmatig gegevens- en funktiemodellen ontworpen. Er zijn richtlijnen voor de projektaanpak.
  - M3: Er is gekozen voor een van de bestaande systeemontwikkelingsmethoden of een eigen aanpak, die goed is gedokumenteerd en vastgelegd. Er wordt gebruik gemaakt van geautomatiseerde hulpmiddelen zoals een data dictionary, er zijn standaards ontwikkeld voor de diverse terreinen van de systeemontwikkeling.
- Computerverspreiding
  - C1: Een centraal mainframe met lokale beeldschermen en/of printers.
  - C1N: Idem, maar met een netwerk voor remote terminals.
  - C2 : Verscheidene rekencentra met lokale of remote terminals.
  - C2N: Idem, maar dan gekoppeld via een netwerk.
  - C3 : Een minicomputer.
  - C3N: Idem, maar dan met een netwerk voor remote terminals.
  - C4 : Een aantal micro-computers.
  - C4N: Idem, maar dan gekoppeld via een netwerk.
  - C5 : Een kombinatie van mainframe en micro`s of mainframe en mini`s en micro`s.
  - C5N: Idem, maar dan onderling gekoppeld.
- Soort projekt.
  - P1 : Het opzetten van een automatiseringsplan met een globaal netwerkontwerp.
  - P2 : Netwerkontwerp op basis van gegevensdistributie.
  - P3 : Nieuwe toepassingen op bestaande systemen.
  - P4 : Nieuwe toepassingen op nieuwe hardware.
  - P5 : Nieuwe toepassingen in verband met de overgang van situatie Cx naar CxN.
  - P6 : Overgang batchverwerking naar interaktieve toepassingen.

- P7 : Evaluatie van bestaande systemen.

Op basis van deze drie invalshoeken wordt het mogelijk de omgeving enigzins in kaart te brengen. Bij de behandeling van deze methoden zal bovenstaande kodering gebruikt worden om het soort omgeving aan te geven. Om wat gevoel te krijgen voor de kodering volgen nu nog enkele voorbeelden.

- Situatie 1: een rekencentrum met een mainframe voor batchverwerking. Men wil een deel van de batchverwerking vervangen door interaktieve toepassingen voor lokale gebruikers. De ontwerpers hebben alleen batch-ervaring, en plegen eigenlijk alleen nog onderhoud. Omgeving: M1-C1-P6.
- Situatie 2: Een centraal rekencentrum met een grote minicomputer en decentraal een aantal stand alone mini's. Er moet een aantal nieuwe toepassingen ontwikkeld worden waarbij gebruik gemaakt wordt van centraal beschikbare gegevens. De automatiseringsafdeling bestaat grotendeels uit mensen die werken in de operationsfeer. De decentrale mini's worden bediend door daarvoor opgeleide gebruikers. Omgeving: M1-C3-P5.
- Situatie 3: een centrale minicomputer voor de administratie. Men overweegt alle vestigingen (25 in Nederland, 3 in België) uit te rusten met een minicomputer voor de lokale administratie. De financiële administratie moet centraal plaatsvinden. Iedere geplaatste mini moet daarvoor gekoppeld worden aan de centrale minicomputer. Er wordt een kostenplaatje van het netwerk gevraagd. De ontwikkelingsafdeling bestaat uit programmeurs, enkele systeemontwerpers en een informatie-analist. In de loop der jaren zijn wat eigen methoden ontwikkeld voor dokumentatie en systeemontwikkeling. Omgeving: M2-C3N-P4.
- Situatie 4: een centraal rekencentrum met een uitgebreide, zeer deskundige automatiseringsstaf. Men gebruikt de modernste hulpmiddelen bij de ontwikkeling van systemen, maar netwerkontwerp is een zaak voor technici. Een aantal vestigingen wordt voorzien van een eigen computer. Gegevens worden centraal ontworpen, decentraal op verschillende plaatsen gebruikt. Men wil de lokatie van gegevens af laten hangen van de kosten voor het netwerk en de te verwachten responsetijden. Omgeving: M3-C3N-P2.
- Situatie 5: een klein bedrijf wil een minicomputer aanschaffen voor lokale gebruikers. Een van de moederbedrijven beschikt over een kleine automatiseringsafdeling. Men vraagt zich af hoe groot de computer moet worden. Omgeving: M1/M2-C3-P4.
- Situatie 6: in een gedecentraliseerd bedrijf staan enkele stand alone minicomputers die niet met elkaar kunnen kommuniceren en op ad hoc-basis zijn aangeschaft. De direktie wil een automatiseringsplan op tafel hebben waarmee richting kan worden gegeven aan verdere automatisering. Omgeving: M1/M2-C3N-P1.
- Situatie 7: op een twee jaar oud mainframe draaien enkele toe-

passingen voor lokale gebruikers. De responsetijden zijn voor de meeste gebruikers niet akseptabel. Er moeten nog diverse projekten worden gerealiseerd. De ontwerpers vragen zich af hoe die responsetijden ontstaan. Omgeving: M1/M2-C1-P7.
Natuurlijk kan het in de tekst van de volgende hoofdstukken ook gaan om een aspekt van een omgeving. Zo kan er sprake zijn van een C1-omgeving of een P2-situatie of van een M3-Cx-omgeving. In het laatste voorbeeld wordt dus een omgeving bedoeld waarin geen netwerk voorkomt, zonder dat de computerverspreiding van belang is. De C1-omgeving zegt niets over de grootte van het bedrijf of het gebruik van methoden en middelen. Alle andere verschillen tussen bedrijven zijn niet van belang voor de te behandelen methoden.
Als men vindt dat een vierde-generatietaal een gebruikerstaal is, dan kan men de verwerking op een transaktieschema natuurlijk beschrijven in die taal. In dat geval blijft het transaktieschema immers een gebruikersdokument. Zie Fig. 41.3 naar aanleiding van voorbeeld 7 op pagina 234 van (45).

Transaktieschema centraal

Transaktienaam: Leverancier per produkt

Menselijke handelingen	Transport	Machinale verwerking
		Display scherm S1
	(----	
Type het produkt nummer in		
	----)	Lees P2 Selekt unique lev.naam from selectie, leverancier where selectie-lev.no and selectie-prod.no = 'P2' Display leverancier
	(----	

Fig. 41.3 Transaktieschema en vierde-generatietaal.

Hoofdstuk 42
Transakties

42.1 Transaktie als entiteitstype

Het begrip transaktie heeft niets te maken met het begrip transaction in de engelstalige literatuur. Een transaction is een aanduiding voor een aktie op een database of, wat algemener, de verwerking door de computer die volgt op het indrukken van een ENTER-toets. Een transaction is dus een begrip van en voor automatiseerders. Transaction analysis is de analyse van de verwerking door programmamodules. Het begrip transaktie is primair een gebruikersbegrip waar automatiseerders mee kunnen werken. Voor de gebruiker is een transaktie de beeldschermversie van een bestaande handmatige procedure. Hoe het beeld van een transaktie ontstaat wordt in de volgende paragraaf behandeld.
In het gemiddelde bedrijf zijn de interaktieve toepassingen bij de automatiseerders alleen bekend in termen van de bekende, aloude begrippen programma`s en bestanden. Daar is immers ook de meeste tijd in gestoken, zowel in het ontwerp als in de bouw. Heel duidelijk wordt dat wanneer er problemen ontstaan. Er is een konfiguratie aangeschaft, nog niet de helft van de toepassingen draait en het systeem zit al vast. Dan blijkt niemand te weten hoe het zit met de bezetting van de beeldschermen, de zwaarte van de toepassingen, de pieken. Een ander voorbeeld is de situatie

waarin twee konfiguraties moeten fungeren als elkaars back-up. Het is niet waarschijnlijk dat de over blijvende machine probleemloos de belasting van de andere machine erbij kan hebben. Er moet dus onderzocht worden in hoeverre de machines belast worden door de beeldschermtoepassingen en hoeveel reservekapaciteit er beschikbaar is voor de uitwijk. Daar blijken geen gegevens over te bestaan: iedereen heeft zich bezig gehouden met de kwalitatieve aspekten van de automatisering en niet met de kwantitatieve aspekten. Die worden hoogstens meegenomen in een paar bekende trajekten zoals het aantal toepassingspaden tot de database en het aantal stekkers voor beeldschermen. Dat laatste trouwens alleen, omdat de computerleverancier nu eenmaal een konfiguratie moet samenstellen.
Zoals ieder entiteitstype bij het gegevensontwerp nauwkeurig in kaart gebracht wordt, zo moeten ook transakties ontworpen worden met hun attributen. Onderstaand is een minimum aantal attributen aangegeven, gebaseerd op de methoden die worden toegepast om transakties te ontwerpen. Uiteraard kan het aantal naar behoefte worden uitgebreid. Het gaat er niet om een gegevensmodel te ontwerpen zoals dat bij data analyse gebeurt, ook al doet het woord "entiteitstype" dat vermoeden. Het gaat om het vastleggen van gegevens van transakties bijvoorbeeld in een data dictionary. De attributen zijn gesplitst in vier soorten: algemene, ergonomische, technische en netwerkattributen. Onder algemene vallen de aspekten die niet duidelijk in een van de andere soorten thuis horen. Tot de ergonomische attributen behoren alle zaken die de gebruikers betreffen. Met technische attributen worden aspekten bedoeld die iets te maken hebben met de konfiguratie, de belasting en de responsetijden. De netwerkattributen betreffen gegevens die nodig zijn om een netwerk te ontwerpen. In Cx-omgevingen zullen deze attributen dus niet voorkomen, in CxN-omgevingen worden ze gebruikt voor het ontwerpen van het netwerk.
Entiteitstype: Transaktie

- Algemene attributen
  - Transaktienaam
  - Status
  - Schermen
  - Transaktiedefinitie
  - Lokatie(s)
  - Printer (J/N)
- Ergonomische attributen
  - Frequentie
  - Pieken
  - Terminaltransaktietijd (T.T.T.), tijdens dialoogsimulatie
    - Totaal
    - Intiktijd

  - Aan- en uitlooptijd
  - Denk- en wachttijd
  - Printtijd
  - Overige
- Terminaltransaktietijd (T.T.T.), berekend met Transaktie analyse:
  - Totaal
  - Intiktijd
  - Aan- en uitlooptijd
  - Denk- en wachttijd
  - Printtijd
  - Overige
- Beeldschermuren per dag
- Aantal beeldschermen
- Aantal printers
- Lokatie van printer(s)
- Gegevensgebruik

- Technische attributen
  - Response-eenheden per transaktie
  - Response-eenheden per seconde
  - ENTER's per uur
  - Gemiddelde dialoogresponsetijd
  - Gemiddelde afsluitresponsetijd
  - Printregels per uur
- Netwerkattributen
  - Gemiddelde berichtlengte heen
  - Gemiddelde berichtlengte terug
  - Invoerrepetitietijd ongunstig
  - Gemiddelde dialoogresponsetijd
  - Gemiddelde afsluitresponsetijd
  - Kommunikatie met andere systemen
    - Batch (J/N)
    - Interaktief (J/N)

De technische en de netwerkattributen worden behandeld in het deel voor de transaktie-analisten. We zullen nu de algemene en ergonomische attributen per stuk bespreken.

- Transaktienaam. Deze naam kan het beste gekozen worden door de gebruiker tijdens het gesprek waarin het transaktieschema wordt gemaakt. In de kop van het transaktieschema wordt die naam dan ook vermeld. De transaktie-analist zal dezelfde naam gebruiken bij Transaktie analyse, zodat de naam verschijnt als kopregel van elk pagina output van het rekenprogramma. Wanneer de transaktie als entiteitstype wordt opgenomen in een data dictionary kan de toegang voor wijzigingen van de attributen verdeeld worden tussen informatie-analist en transaktie-analist. De maximale lengte voor een transaktienaam is, wat het rekenprogramma betreft, 32 tekens.

- Status. In dit attribuut kan bijvoorbeeld met een cijfer de status van het ontwerp van de transaktie worden aangegeven.
Bijvoorbeeld:
1: Naam vastgesteld dd:
2: Transaktieschema gereed dd:
3: Dialoogsimulatie voorbereid dd:
4: Eerste keer dialoogsimulatie uitgevoerd dd:
5: Dialoogsimulaties dd: , , ,
6: Dialoogsimulatie geaksepteerd dd:
7: Definitief transaktieschema gereed dd:
8: Transaktieschema overgedragen voor Transaktie analyse dd:
9: Resultaten ergonomische transaktie analyse verwerkt dd:
10: Transaktie ontwerp geaksepteerd dd:
11: Logische Transaktie analyse uitgevoerd dd:
12: Transaktie ontwerp geaksepteerd dd:
13: Technische Transaktie analyse uitgevoerd dd:
14: Transaktie ontwerp geaksepteerd dd:
In dit voorbeeld geven 10 en 12 de mogelijke iteraties aan.
- Schermen. Tijdens dialoogsimulatie moeten namen worden toegekend aan schermen. Het is natuurlijk verstandig die namen zoveel mogelijk te laten aansluiten op standaards en definitieve namen. In ieder geval worden hier de schermen genoemd die deel uitmaken van deze transaktie.
- Transaktiedefinitie. Hier wordt de naam van de transaktie op de dialoogsimulator aangegeven. Hier kan natuurlijk ook de transaktiedefinitie worden vastgelegd.
- Lokatie(s). In Cx-omgevingen is dit attribuut voor het ontwerpproces niet direkt noodzakelijk. Wanneer echter de kans bestaat op een overgang van Cx naar CxN, is het verstandig hier aan te geven op welke werkplekken een transaktie wordt gebruikt. Wanneer bij de gegevens wordt vastgelegd in welke transaktie ze worden gebruikt, is via dit attribuut, zeker in M3-omgevingen, gemakkelijk vast te stellen waar welke gegevens worden gebruikt. Wanneer vervolgens een lokatie is gekozen voor de opslag van de gegevens, kan de hoeveelheid verkeer in een CxN-omgeving worden bepaald. Deze aspekten worden in het deel voor de transaktie-analisten verder uitgewerkt.
- Printer. Met dit attribuut wordt aangegeven of er ten dienste van de transaktie geprint wordt. Wanneer het printen deel uitgemaakt van de transaktie, dan is het printen aangegeven op het transaktieschema en daarmee opgenomen in de terminaltransaktietijd. Het kan ook zijn dat het printen volledig gescheiden is van het werken aan de beeldschermen, bijvoorbeeld omdat aan het eind van de dag een batchprintjob wordt gestart. In dat geval wordt de benodigde printtijd niet via Transaktie analyse bepaald. In het eerste geval zijn twee situaties mogelijk: een printer per beeld-

scherm of een printer voor verscheidene beeldschermen. Bij de analyse van het printwerk gaat het meestal om een van de volgende aspekten: het aantal benodigde printers, de gevraagde printkapaciteit per transaktie, de printtijd ten opzichte van de terminaltransaktietijd. In de paragraaf Resultaten en konklusies van Transaktie analyse werken we dit verder uit.
- Frequentie. Dit attribuut wordt ingevuld tijdens de analysefase of tijdens het transaktie-ontwerp. Het is uiteraard een gegeven dat van de gebruikers komt, want transakties moeten entiteiten zijn die de gebruiker kent. De frequentie kan aangegeven worden in aantal transakties per dag, per week, per maand of iets dergelijks. Bijvoorbeeld: gemiddeld 200 per dag, maximaal 250.
- Pieken. Ook dit is een gebruikersgegeven, dat in free format wordt opgenomen. De terminologie moet ook voor de transaktie-analist duidelijk zijn. Er kan bijvoorbeeld staan: vrijdags 300 per dag.
- T.T.T. Dit gegeven is een resultaat van Transaktie analyse. Afhankelijk van de situatie wordt hier de bruto of de netto T.T.T. overgenomen uit het resultaten overzicht van Transaktie analyse. De terminaltransaktietijd (T.T.T.) geeft aan hoe lang een transaktie duurt. Dit gegeven wordt onder andere gebruikt bij de berekening van een aantal ergonomische en technische attributen.
De onderverdeling van de T.T.T. wordt overgenomen uit de resultaten van Transaktie analyse. Deze verdeling van de T.T.T. geeft een snel inzicht in de soort transaktie: high speed data entry, veel of weinig handmatig werk, hoeveelheid printwerk. Deze attributen geven richting aan het zoeken naar alternatieve oplossingen. Wanneer bijvoorbeeld het aantal terminals of de bezetting ervan teruggebracht moet worden, kunnen eerst de transakties met grote aan- en uitlooptijd of denk- en wachttijd onder de loep genomen worden. Bij transakties die voor 90% bestaan uit intiktijd valt in dit opzicht waarschijnlijk weinig te verdienen.
- Uren per dag. Deze tijd geeft aan hoeveel uur per dag er in totaal nodig is om het aantal transakties per dag uit te voeren. Dit attribuut kan deel uitmaken van de sociale aspekten van de automatisering. Het geeft tijdens het logisch ontwerp inzicht in de bezetting van medewerkers, het aantal beeldschermen en daarmee in de belasting van het systeem.
- Aantal beeldschermen. Dit gegeven kan bepaald zijn zonder gebruik te maken van Transaktie analyse. Wanneer een manager, onafhankelijk van het geringe gebruik ervan, op ieder bureau een beeldscherm wil hebben, wordt dit attribuut bepaald door het aantal bureaus op zijn afdeling. Vanzelfsprekend hebben het aantal beeldschermen en de transakties die er op uitgevoerd kunnen worden, gevolgen voor de belasting van het systeem. Wanneer het aantal beeldschermen wordt bepaald op basis van het aantal uit te

voeren transakties per dag wordt dit attribuut berekend door de informatie-analist of de transaktie-analist. In het algemeen wordt als basis dan een aantal verschillende transakties genomen. Per transaktie wordt met dit attribuut aangegeven op hoeveel beeldschermen de transaktie kan worden uitgevoerd. De tijdverdeling van een transaktie over de verschillende beeldschermen wordt op het bezettingsoverzicht weergegeven.
- Aantal printers. Ook dit attribuut kan bepaald zijn door iets anders dan de resultaten van Transaktie analyse, omdat bepaalde dokumenten op een bepaalde plaats geprint moeten worden. Het aantal printers kan ook door de informatie-analist of de transaktie-analist bepaald zijn door berekening van de benodigde kapaciteit. Dit attribuut geeft aan op hoeveel printers er ten dienste van de transaktie geprint wordt. Wanneer verschillende beeldschermen gebruik maken van een printer kan het aantal kleiner zijn dan 1.
- Lokatie van printers. Meestal alleen van belang in CxN-omgevingen in verband met het transport van printregels via het netwerk. Wanneer het gaat om een situatie met veel lokaties met printers kan het om ergonomische redenen echter ook in Cx-omgevingen nuttig zijn om een overzicht te maken van beeldschermlokaties, printlokaties en bijbehorende transakties. Met name de gebruikers hebben belangstelling voor dergelijke overzichten, liefst lange tijd voor de invoering, bijvoorbeeld tijdens het logisch ontwerp.
- Gegevensgebruik. Hier kan worden vastgelegd welke gegevens er in een transaktie worden gebruikt. Er kan worden gerefereerd aan items, entiteitstypes, bestanden of subschema`s. Met name in distributieve omgevingen is het van belang voor het netwerkontwerp vast te leggen in welke transakties welke gegevensverzamelingen worden gebruikt.
Onder de algemene attributen kan ook een indikatie worden opgenomen over de onmisbaarheid van de transaktie gedurende een uitwijksituatie. Zo kan bijvoorbeeld het aantal uren of dagen dat de transaktie onbruikbaar mag zijn, worden aangegeven. Systeembelastings- en verkeersattributen maken het mogelijk de uitwijksituatie te ontwerpen.
In Fig. 42.1 is de transaktie als entiteitstype weergegeven als onderdeel van een gegevensmodel. Als dit gegevensmodel wordt opgenomen in een datadictionary dan kan men er transaktieschema`s mee onderhouden, men kan per transaktie opvragen welke schermen erbij horen, welke processen er in voorkomen, welke funktietoetsen gebruikt worden enzovoort.
In het deel voor de transaktie-analist is het gegevensmodel nog uitgebreid met de geografische situatie. Dit model geeft uitstekend weer wat een transaktie eigenlijk is en wat de relatie is met andere aspekten van interaktieve toepassingen. We zullen daarom de blokken in Fig. 42.1 kort toelichten.

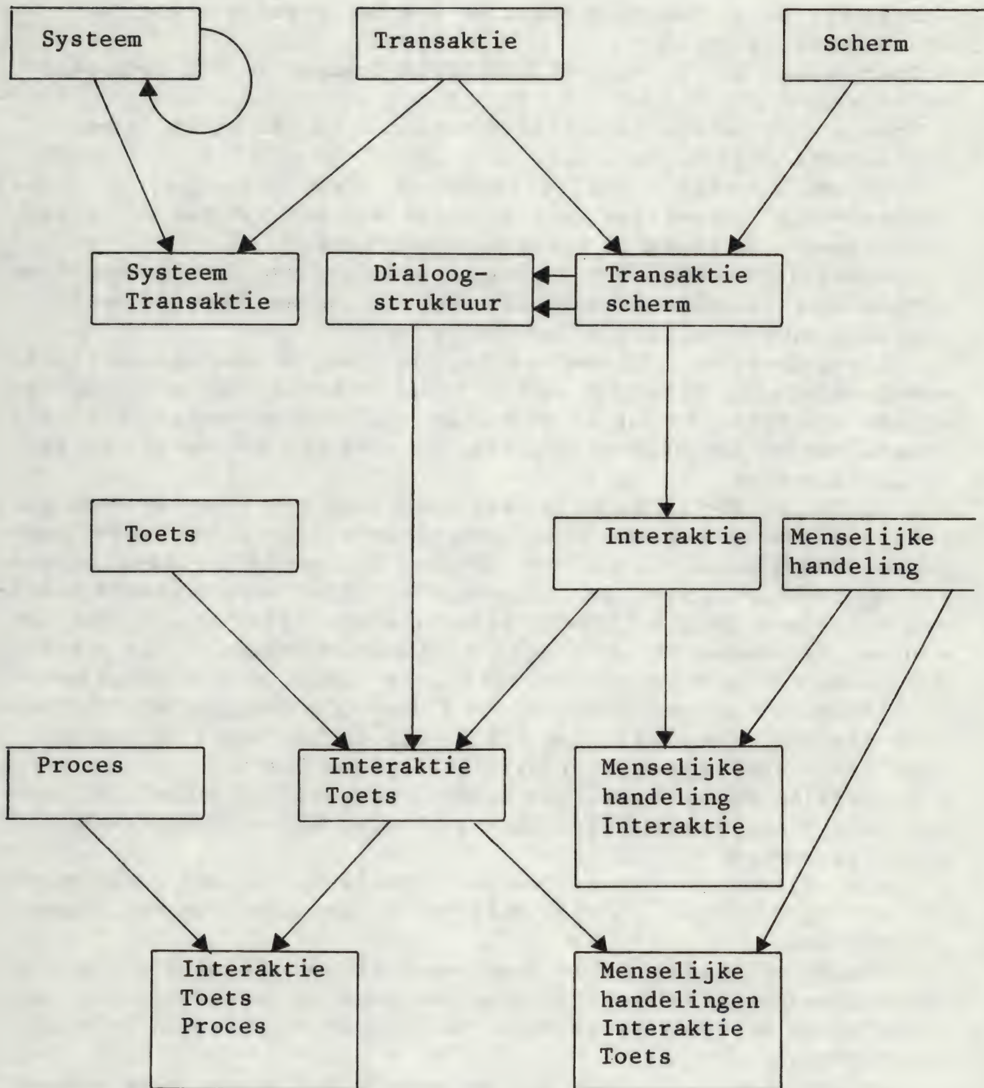


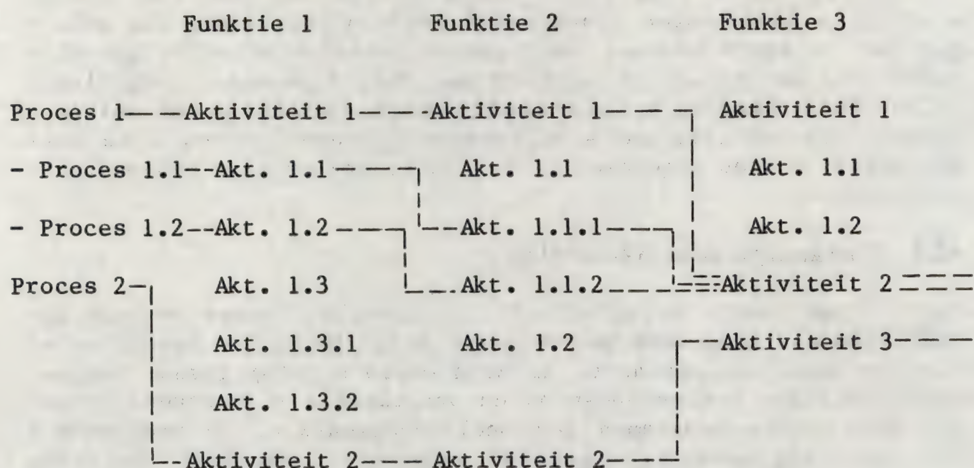
Fig. 42.1 De transaktie als onderdeel van een gegevensmodel.

- Systeem. Het hele informatieverwerkende systeem bestaat natuurlijk uit een groot aantal delen: software, bestanden, hardware enzovoort. In dit model brengen we van dat systeem alleen de transakties in beeld.
- Transaktie. Dit is het entiteitstype Transaktie met uiteraard de besproken attributen.
- Scherm. Dit zouden de schermen kunnen zijn zoals die in een bibliotheek zijn opgenomen.
- Systeem Transaktie. Dit relatieblok maakt het mogelijk op te vragen welke transakties deel uitmaken van een systeem of in welke systemen een bepaalde transaktie voorkomt.
- Transaktie Scherm. Dit relatieblok zorgt voor het antwoord op vragen als: in welke transakties komt dit scherm voor of welke schermen maken deel uit van deze transaktie.
- Dialoogstruktuur. In een dialoogstruktuur is aangegeven hoe de schermen elkaar opvolgen en op basis waarvan. Op een dialoogstruktuurdiagram is dus de struktuur en de flow weergegeven. Per scherm kan nu gevraagd worden naar de naam van het vorige en het volgende scherm.
- Interaktie. Dit is de pijl naar rechts op een transaktieschema. Iedere interaktie wordt voorafgegaan door menselijke handelingen en wordt gerealiseerd met het drukken op een ENTER-toets of een van de funktietoetsen. Per transakties kunnen verschillende schermen voorkomen en per scherm verschillende interakties. Men zou binnen een transaktie de interakties moeten nummeren. In dit model kunnen twee soorten interaktie van elkaar worden onderscheiden: interakties per scherm (via Transaktie Scherm) en interakties die een schermwisseling tot gevolg hebben zoals is vastgelegd in de dialoogstruktuur (via Interaktie Toets).
- Menselijke handelingen. Een beschrijving van de menselijke handelingen zoals die per pijl naar rechts op het transaktieschema zijn beschreven.
- Toets. Hiermee worden de toetsen bedoeld die op het toetsenbord worden ingedrukt en een pijl naar rechts betekenen op het transaktieschema.
- Interaktie Toets. Nu is het mogelijk om per scherm vast te stellen welke interakties er plaats vinden en per interaktie: met welke toets dat gebeurt en welke menselijke handelingen er aan voorafgaan.
- Interaktie Toets Proces. Per interaktie kan nu gevraagd worden naar de processen die erbij horen.
- Proces. Hier kan men algoritmes, programma's of funkties opnemen zoals die op transaktieschema's in de rechterkolom voor kunnen komen.
- Menselijke handelingen Interaktie. Per interaktie zijn de menselijke handelingen nu op te vragen.

- Menselijke handelingen Interaktie Toets. Met dit blok kunnen menselijke handelingen voorafgaand aan een interaktie nog afhangen van de toets waarmee een bepaalde interaktie wordt gerealiseerd. Het zal duidelijk zijn dat een dergelijk model, opgeslagen in een data dictionary een krachtig hulpmiddel is in het ontwerpproces. Transaktieschema's bijvoorbeeld kunnen nu met elke mate van detail worden opgenomen in de dictionary en zijn eenvoudig te onderhouden.

42.2 Van analyse naar transakties

Zoals in de vorige paragraaf al is opgemerkt, vormen de interaktieve toepassingen vaak een ondoorzichtige brij, die kwantitatief niet in kaart is gebracht. Er zijn massa's metagegevens vastgelegd, relaties tussen gegevens en programma's zijn bekend, evenals die tussen programma een beeldschermmasker, de programma's zijn prachtig modulair opgebouwd enzovoort, enzovoort. De vraag op hoeveel beeldschermen een toepassing gebruikt wordt kan uit de ontwerpdokumentatie vaak al niet meer worden beantwoord: dat is immers niet van belang voor de logica van programma's en gegevensverzamelingen. De vraag naar de "zwaarste" transaktie wordt nooit beantwoord. Bovendien weet niemand hoe die zwaarte bepaald zou moeten worden. Als ontwerpers het hebben over zware toepassingen, bedoelen ze programma's met zware database-verwerking. En dat in een wereld waarin bijna iedere ontwerper al eens te maken heeft gehad met lange responsetijden ten gevolge van overbelaste systemen.
Het voordeel van bestaande systeemontwikkelingsmethoden is het feit dat de werkwijze tijdens analyse en ontwerp is beschreven. Bij een bestudering van veel van die methoden en van de opleiding voor informatie-analisten en systeemontwerpers blijkt dat niets aanmoedigt tot het ontwerpen van transakties, ondanks het gebruik van termen als transactions en transacties. De handboeken voor ontwerpers staan bol van termen als dialoogontwerp, mens-machinedialoog, schermontwerp, interaktiemodellen, konversatiediagrammen, dialoogstruktuurdiagrammen en vele andere termen, die aangeven dat er interaktieve toepassingen worden ontworpen naast batchapplikaties en handmatige aktiviteiten. Eenzelfde ondoorzichtige brij komen we tegen in de analysefase. Taak/funktie-omschrijvingen blinken meestal niet uit in het gestruktureerd opsommen van alle aktiviteiten en procedures. Nu is de taak van een medewerker op een verkoopkantoor natuurlijk ook komplex in vergelijking met het werk van iemand aan een lopende band. In de analysefase proberen we toch de funktie van medewerkers af te breken naar aktiviteiten, procedures en handelingen. Pas wanneer we dat in vol-



Akt. 1.3.1	Akt. 1.3.2
- Naam:	- Naam:
- Doel:	- Doel:
- Procedure:	- Procedure:

Akt. 1.3 - Tr.aktie 1.3? Doel:

Akt. 1.3.1 - Tr.aktie 1.3.1? Doel:

Akt. 1.3.2 - Tr.aktie 1.3.2? Doel:

Fig. 42.2 Aktiviteiten en transakties.

doende mate hebben gedaan, kunnen we gaan praten over het inzetten van een beeldscherm en over transakties. Dan worden er transakties ontworpen zoals in het hoofdstuk Transakties is beschreven en komen er bruikbare cijfers beschikbaar. De vraag is nu: hoe gedetailleerd moet de analyse zijn om transakties te kunnen herkennen? Bij de beantwoording gaan we ervan uit dat de informatie-analist ervaring heeft met interaktieve toepassingen op welke manier dan ook, omdat anders de herkenning niet tot stand komt. Een informatie-analist die zich een bepaalde aktiviteit niet op een beeldscherm kan voorstellen raakt in meer dan een opzicht in de problemen. Door gebrek aan standaardisatie in het gebruik van termen is het niet mogelijk de diepgang van de analyse met een woord aan te geven. In de diverse systeemontwikkelingsmethoden worden veel begrippen gehanteerd om handelingen of werkzaamheden aan te geven. Om misverstanden te voorkomen is in Fig. 42.2 aangegeven wat er met de verschillende begrippen wordt bedoeld.
Een bedrijf wordt opgesplitst in funkties: verkoop, produktie, inkoop etc. Per funktie worden een aantal aktiviteiten onderscheiden. Binnen verkoop bijvoorbeeld het maken van offertes, het behandelen van reparatie-opdrachten, het behandelen van reklamaties over levertijden, orderverwerking.
Een aktiviteit kan weer gesplitst worden in reeks aktiviteiten. zie Fig. 42.2. Aktiviteit 1 wordt bijvoorbeeld in aktiviteiten 1.1, 1.2 en 1.3 gesplitst, aktiviteit 1.3 in 1.3.1 en 1.3.2. Karakteristiek voor een aktiviteit zijn: naam, doel, methode en procedure. Op een bepaald moment gaat de detaillering zover, dat beter de procedure beschreven kan worden, dan dat een volgende splitsing in aktiviteiten plaatsvindt. In feite wordt dan de aktiviteit gekarakteriseerd door de procedure te beschrijven. Het is duidelijk dat "Het pakken van een pen en een orderbon" bij het aksepteren van een order een te fijne onderverdeling is, maar "Orderverwerking" kan weer te grof zijn. Als we namelijk vragen naar de procedure rond de orderverwerking kan bijvoorbeeld blijken dat er drie soorten orders zijn die elk een totaal andere afhandeling vragen. Dan zijn er dus drie procedures en de procedure orderverwerking is blijkbaar niet interessant. Er kunnen drie aktiviteiten onderscheiden worden: levering uit voorraad, levering na gereed komen van voorraadartikel en levering van maatwerkartikel. Laten we als voorbeeld nemen levering van maatwerk. Deze aktiviteit kan gesplitst worden in: kontrole of er een offerte is gemaakt, offerte vergelijken met opdracht, kontrole aanwezige grondstoffen, bepalen van de levertijd op basis van de produktieplanning.
Bij iedere fase in de detaillering dient de informatie-analist zich af te vragen of hij zich een aktiviteit als een transaktie kan voorstellen. Als hij dat op het niveau orderverwerking zou

proberen, dan blijkt dat al niet te gaan, omdat daar meteen drie totaal verschillende procedures onder vallen. Een niveau lager staan enkele aktiviteiten die best met een beeldscherm gerealiseerd kunnen worden. Kontrole of er een offerte voor de klant is gemaakt betekent het intypen van een klantnummer of een zoekargument. Wanneer de offerte zelf ook is opgeslagen kan deze aktiviteit uitstekend gekombineerd worden met de volgende aktiviteit: vergelijking van de offerte met de opdracht. Dat zou dus een transaktie OPVRAGEN OFFERTE kunnen zijn. De volgende aktiviteiten: kontrole aanwezige grondstoffen betekent het intypen van een grondstofaanduiding en het lezen van de aanwezige voorraad. De handeling moet misschien enige malen herhaald worden. De kontrole op aanwezigheid van grondstoffen zou dus de transaktie OPVRAGEN VOORRAAD GRONDSTOFFEN kunnen zijn. Op soortgelijke wijze zou een transaktie BEPALING LEVERTIJD bedacht kunnen worden. Dit zijn transakties die iedere analist zich op een beeldscherm moet kunnen voorstellen. De vraag is nu: wat gaat er fout wanneer de analist een niveau te vroeg of een niveau te laat probeert van een aktiviteit een transaktie te maken?
In het eerste geval probeert hij zich een transaktie MAATWERKOPDRACHT voor te stellen. De gebruiker typt het klantnummer in en de offerte wordt gedisplayd. Wanneer die akkoord is bevonden, typt hij de zoekargumenten voor de grondstoffen in en het systeem toont de voorraden. Als alles beschikbaar is gaat hij met een funktietoets bijvoorbeeld, door naar de produktieplanning. Er is niets fout aan deze transaktie in vergelijking met een ontwerp dat drie transakties voorstelt. Er is wel een duidelijk verschil. In het eerste geval is de gebruiker veel vrijer in de manier van werken. Hij kan eerst van alle orders de offertes kontroleren, vervolgens alle grondstoffen en vervolgens de produktieplanning. In het tweede geval moet een order in z'n geheel worden afgehandeld. De ene gebruiker zal direkt uit deze alternatieven kiezen, de andere zal het pas weloverwogen kunnen, als hij beide heeft uitgeprobeerd op de dialoogsimulator. In het hoofdstuk Dialoogsimulatie is opgemerkt dat er minstens een alternatief gemaakt moet worden van een startontwerp. Informatie-analisten die moeite hebben met de overgang tussen analyse en ontwerp zullen dus per definitie gaarne enige alternatieven uitwerken!
Als de analist een niveau te laat denkt aan het ontwerpen van een transaktie komt hij al snel terecht bij aktiviteiten die niet meer zijn voor te stellen op een beeldscherm. Laten we als voorbeeld nemen het kontroleren of er een offerte is gemaakt. De gebruiker loopt naar de dossierkast OFFERTES, zoekt de offerte op en neemt die mee naar zijn bureau. Hij leest de tekst en vergelijkt die met de opdracht. Geen van deze aktiviteiten is op zich te vertalen naar een transaktie. Wanneer er op een scherm iets

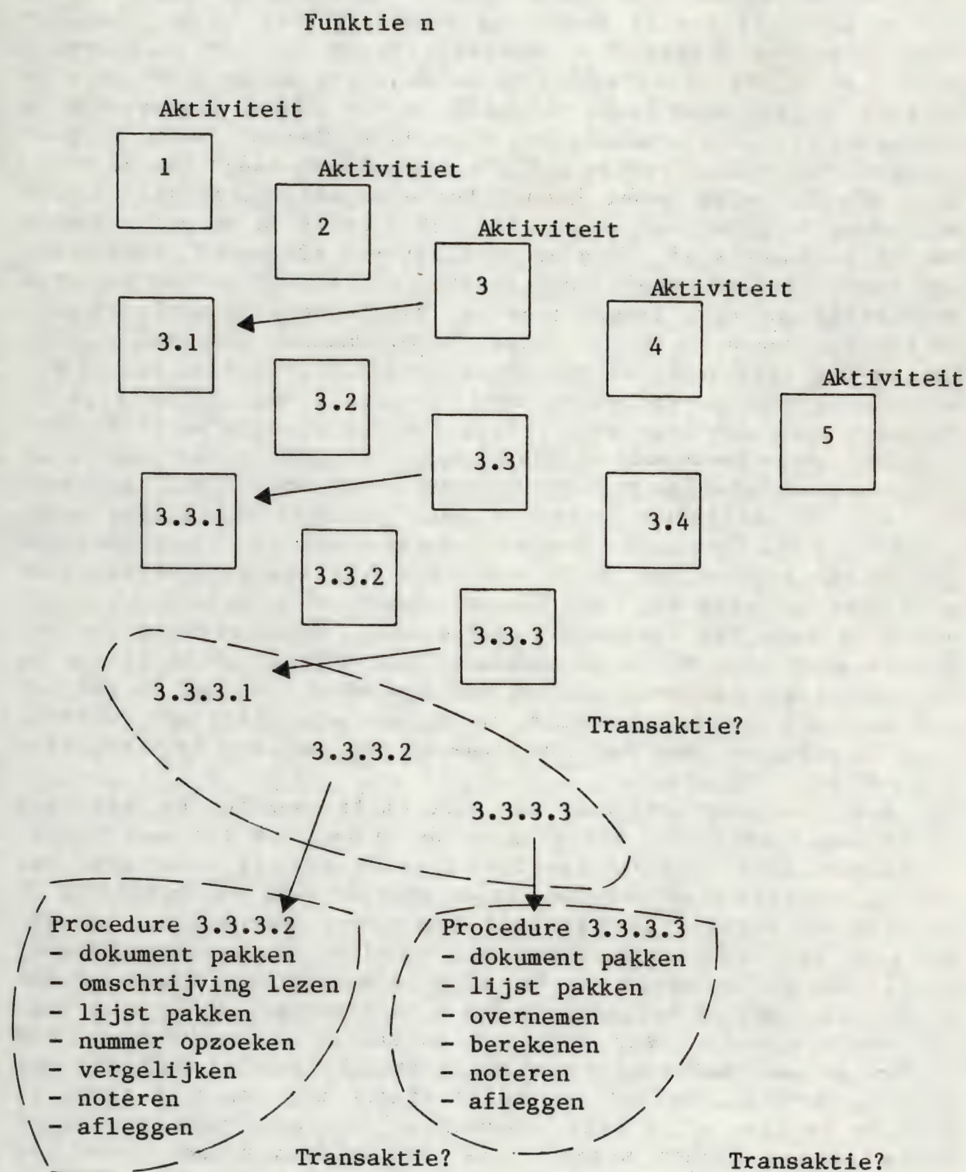


Fig. 42.3 Detaillering in de analysefase.

gelezen wordt, moet daar het intypen van iets aan vooraf zijn gegaan. De transaktie OFFERTE LEZEN is onmogelijk. Samengevat stellen we dus vast dat er vaak geen eenduidig punt in de detaillering is waarop transakties ontstaan. Er is wel een punt waarop duidelijk is dat de detaillering te fijn is. Wanneer dat punt is bereikt, is een stap terug voldoende om een aantal transakties te ontwerpen. Of die transakties, door twee stappen terug te gaan gekombineerd kunnen worden tot één transaktie, hangt van de situatie af. In ieder geval kunnen beide mogelijkheden gesimuleerd worden en de gebruiker kiest. Fig. 42.3 geeft de aanpak nog eens in het algemeen weer. We gaan even uit van N2-charts. Aktiviteiten worden onderverdeeld in aktiviteiten tot het moment dat verdere verdeling niet zinvol meer is. Vanaf dat punt worden procedures beschreven in een of andere vorm. Wanneer dat punt bereikt is, is het verstandig te denken aan transakties. Soms kan de beschrijving van de procedure heel duidelijk aanleiding zijn te vertalen naar een transaktie. Niet dat die transaktie in de analysefase ontworpen moet worden, maar de analist kan zich deze procedure voorstellen als een transaktie. Zo worden alle procedures van de aktiviteit onder de loep genomen. Vervolgens wordt gekeken of de kombinatie van procedures misschien als transaktie gezien kan worden: dan wordt een aktiviteit een transaktie. Voor alle zekerheid kan nog even bekeken worden of de hele groep aktiviteiten soms een transaktie kan worden. Belangrijk is daarbij hoe de gebruiker er tegenaan kijkt. Een belangrijk kriterium is de herhaling. Een aktiviteit of een procedure die door de gebruiker herhaald wordt uitgevoerd, wordt waarschijnlijk een transaktie. De gebruiker kan dan ook aangeven hoe vaak hij de aktiviteiten per dag uitvoert.

Het doel van deze paragraaf was niet om transakties te ontwerpen of te laten ontstaan. Het ging om de vraag: hoe ver moet de detaillering in de analyse gaan? Wanneer de analist zover gaat dat hij transakties ziet ontstaan is de detaillering voldoende. In de ontwerpfase wordt op dat punt de draad weer opgepakt bij het maken van het transaktieschema. Dan ontstaat de geautomatiseerde versie van de procedure. In Fig 42.3 is zonder meer duidelijk dat de aktiviteiten in Procedure 3.3.3.2 niet verder behoeven te worden onderverdeeld. Deze aktiviteiten kunnen zo worden opgenomen in het transaktieschema onder de menselijke handelingen. Als dat punt is bereikt, is het misschien zinvol nog een stap terug te gaan om te zien of de drie procedures later misschien een transaktie zouden kunnen worden. Als dat duidelijk kan, hoeft de analist niet verder te gaan dan het niveau van aktiviteit 3.3.3. In de praktijk kan men bij het ontwerpen van de transakties toch nog wel in moeilijke situaties terecht komen. Order entry kan

Analysefase

Funktie-analyse	Gegevensanalyse
Funktiemodel	Gegevensmodel
Funkties	Entiteittypes
Aktiviteiten n	Attributen

Aktiviteiten n.m.

Aktiviteiten n.m.p. - mogelijke transakties

Procedures - mogelijke transakties

Logisch ontwerp

Aktiviteiten - mogelijke transakties: Transaktie-ontwerp

Procedures - mogelijke transakties: Transaktie-ontwerp

Transaktie-ontwerp = Dialoogsimulatie + Transaktie analyse

Dialoogsimulatie = Transaktieschema maken + dialoog simuleren

Fig. 42.4 Van analyse naar ontwerp

eenvoudig klinken. Er bestaan echter bedrijven waar men zeer uitgebreide, ingewikkelde orders verwerkt. Een voorbeeld daarvan zijn de leveranciers van verwarmings- en airconditioninginstallaties. Er wordt een order ontvangen voor de aangeboden installatie. Een airconditioningsinstallatie bestaat uit een aantal eenheden. Eenheden voor verwarming, bevochtiging, naverwarming, ventilatie, koeling enzovoort. Iedere eenheid op zich bestaat weer uit een aantal eenheden, elk met hun technische specifikaties. Een gebruiker kan best een morgen bezig zijn met het invoeren van een order. Omdat iedere installatie bestaat uit een kombinatie van elementen zal al snel duidelijk zijn dat er per element een transaktie zal worden ontworpen. Het maken van transaktieschema's en het uitvoeren van dialoogsimulatie blijft dus zinvol. Of het uitvoeren van Transaktie analyse ook zinvol is, is de vraag. Het repeterende karakter is nihil, kwantiteiten zijn moeilijk aan te geven, het gaat vaak om een klein aantal orders per jaar, na dialoogsimulatie heeft de gebruiker een indruk van de tijdsbesteding en de verwerking zal bestaan uit korte pieken. Het is moeilijk een algemene regel te geven, maar als Transaktie analyse wordt uitgevoerd moet het doel ervan goed duidelijk zijn. De informatie-analist hoeft alleen maar de ergonomische resultaten van Transaktie analyse te kennen en dus moet hij zich afvragen of een dergelijke analyse voor hem en de gebruikers iets oplevert. De andere resultaten van Transaktie analyse zijn ter beoordeling aan de transaktie-analist. In CxN-omgevingen kunnen er best redenen zijn om Transaktie analyse toch uit te voeren. In het deel voor de transaktie-analist zal behandeld worden bij welke toepassingen en omgevingen Transaktie analyse zinvol is.
Tenslotte nog iets over de volgorde in de projektaanpak. In principe komt ontwerp na analyse. Zoals hierboven is aangegeven levert een analyse met de blik op een ontwerp een goede koppeling tussen beide, maar analyse blijft thuis horen in de analysefase en ontwerp in de ontwerpfase, zie Fig 42.4. De manier van werken wordt dus: analyseren, denkend aan transakties. Hoe letterlijk dat denken moet worden opgevat, is een kwestie van smaak. Een analist mag natuurlijk best tijdens de analysefase bij een aktiviteit vast even de namen van eventueel te ontwerpen transakties vaststellen en noteren als geheugensteun, met een korte toelichting op het doel, zie Fig. 42.2. In ieder geval moet hij zich blijven realiseren dat het gaat om de relatie tussen procedures, transakties en funktionarissen, zie paragraaf 34.7. Het ontwerpen van transakties betekent het ontwerpen van de omgang met gegevens. Zoals in de handmatige situatie niet iedereen toegang heeft tot alle bedrijfsgegevens, mag ook niet iedere funktionaris elke transaktie uitvoeren of ontwerpen. Met name in distributieve

omgevingen moet de gebruiker worden geattendeerd op zaken als eigenaarschap, beheer en gebruik van gegevens.
Als er in een komplexe situatie met de gebruikers bijvoorbeeld tijdens de analysefase aan transaktie-ontwerp begonnen wordt, is er helemaal geen basis voor de vaststelling van transakties. Toch zijn er situaties waarin de aktiviteiten zo simpel zijn, dat het transaktie-ontwerp op elk moment, zelfs tijdens het vooronderzoek, kan plaats hebben. Dat kan omdat transaktie-ontwerp zich bij een ergonomische Transaktie analyse helemaal niet bezig houdt met de verwerking door de computer. Na zo'n transaktie-ontwerp is er in principe dan ook niets bekend over de verwerking. De informatie-analist heeft zich alleen beziggehouden met de dialoog en de menselijke handelingen. Voor de gebruiker is de stap van papier naar beeldscherm klein geweest. De aanpak wordt behandeld in de paragraaf Transaktie-ontwerp in verschillende omgevingen (42.4). Een reden voor die aanpak kan bijvoorbeeld zijn dat er tijdens het vooronderzoek een netwerkkoncept of een gedetailleerd informatieplan moet worden gemaakt. De lokaties van de werkplekken en het gebruik van gegevens, zoals dat kan worden afgeleid uit de dialoog, vormen voor de transaktie-analist een basis om een of meer netwerkkoncepten te ontwikkelen.
Deze paragraaf, resumerend, wordt er eigenlijk gesteld dat, in het kader van interaktieve toepassingen, funktie-analyse zonder ontwerpkennis niet mogelijk is, tenzij men steeds tot in het absurde wil detailleren. Bij gegevensanalyse ligt de zaak misschien wat anders. Zie verder paragraaf 43.5.

42.3 Transaktieschema's

Het transaktieschema is het meest onorthodoxe dokument. Het lijkt niet te passen in systeemontwikkelingsmethoden en informatie-analisten die met een dialoogsimulator werken zijn ineens weer volbloed automatiseerders: ze zien alleen nog schermlay-outs en de achterliggende programma's. Het transaktieschema past in elke ontwikkelingsmethode of die nu data-driven, function-driven of output-driven is. Het transaktieschema is de start van het transaktie-ontwerp en dat past in elke methode. In de volgende paragraaf wordt de aansluiting in de verschillende omgevingen behandeld.
Fig. 42.5 is een voorbeeld van een transaktieschema. De linkerkolom beschrijft de handelingen die verricht moeten worden. Het is belangrijk alles volledig te beschrijven,omdat daarmee de basis gelegd wordt voor de bekende wait- and thinktimes in de vakliteratuur. Automatiseerders zijn helemaal gekoncentreerd op de dialoog, omdat de computer daarop moet reageren. Bij het in kaart

Transaktieschema centraal

Transaktienaam: BALIE VERKOOP

Menselijke handelingen	Transport	Machinale verwerking
Naam vragen aan klant.		
Zoekargument intoetsen	----)	Opzoeken NAW verk.
	(----	
Juiste klant bepalen		
en klant nr. intoetsen	----)	Kontroleren op aanwezigheid
	(----	Menuscherm 401A, le helft
Intoetsen balieverk.	----)	
		2e helft displayen
Intoetsen:	(----	
Mutatiekode	----)	Akkoordregel displayen
	(----	
Intoetsen:		
"akkoord"	----)	Selektie 401B en display
	(----	
Intypen van:		
Funktie	----)	Bep: ordernr., verkoper,
	(----	klantnr.
Klantref.	----)	
		Bep: datum, besteld door,
	(----	datum, verzending verreke-
Besteld door	----)	ning enz. Fakt. adr.
	(----	Akkoordregel displayen
Intoetsen "akkoord"	----)	Verwerking + selektie 401C
	(----	en display
Intoetsen artikel nr.	----)	Zoek beschr, korting,
		prijs voorraad, spec.
Kontroleren art.omschr	(----	prijs
intoetsen aantal	(----	
Enter voor akk.	----)	verwerken regel
	(----	cursor pos.
Event. volgende		
regel of afsl. akk.	----)	selektie 401F

Fig. 42.5 Voorbeeld van een transaktie op een start/stop beeldscherm.

brengen van transakties is alles wat er gebeurt naast het intypen minstens zo belangrijk. Bijvoorbeeld de aan- en uitloop, het bladeren in dokumenten, het kontroleren, het opnemen van de telefoon, het paraferen enzovoort. De belasting die een transaktie vormt voor de computer hangt hiervan af. Hoe meer menselijke handelingen, hoe lichter de transaktie wordt. Daarom moet de linkerhelft zo volledig mogelijk de procedure aan het beeldscherm beschrijven. Iedere interaktie wordt aangegeven met een pijl. Onder een interaktie verstaan we het starten van de verwerking door op de computer op een speciale toets te drukken: ENTER, TRANSMIT, SEND, CR of iets dergelijks.
Per pijl naar rechts wordt in gebruikerstermen beschreven welke funktie de computer nu moet verrichten, welke gegevens daarbij nodig zijn en eventueel hoe de berekening moet worden uitgevoerd. De verwerking door de computer zal altijd uitlopen op het displayen ergens van. Dat kan het verplaatsen van de cursor zijn, het tonen van gegevens of het displayen van het volgende scherm. In feite wordt op het transaktieschema de procedure aan het beeldscherm beschreven inclusief de gekozen dialoog.
Om goede transaktieschema's te maken is het nuttig dat een informatie-analist weet wat er tijdens Transaktie analyse mee gebeurt. Daarom wordt Transaktie analyse in een apart hoofdstuk behandeld voorzover dat voor de informatie-analist van belang is.
We zullen nu de belangrijkste aspekten van het transaktieschema bespreken. Vervolgens komen aan de orde de gedetailleerdheid van transaktieschema's, de complexe situaties en de decentrale transaktieschema's
- De aansluiting op de handmatige procedures.

Tijdens de analysefase zijn de bedrijfsprocessen in kaart gebracht. Per werkplek zijn procedures onderkend. Een deel daarvan wordt geautomatiseerd. In de procesbeschrijvingen kan nu worden aangegeven door welke transakties sommige procedures worden vervangen en op welk transaktieschema de beschrijving is te vinden.
- De beschrijving van de menselijke handelingen.

Automatiseerders hebben eigenlijk alleen interesse in wat computers moeten doen. De gebruikers merken achteraf vanzelf wel wat ze moeten doen of ze moeten het maar opmaken uit het pak schermlay-outs. Op het transaktieschema worden de menselijke handelingen beschreven in gebruikerstermen. Sommige gebruikers zullen na korte tijd zelf hun transaktieschema's willen maken. Wie zou de procedure rond het beeldscherm beter kunnen beschrijven?
- Het transaktieschema als gebruikersdokument.

Alle in systeemontwikkelingmethoden voorkomende dokumenten worden gemaakt door en zijn bestemd voor de automatiseerders. Van sommige dokumenten wordt beweerd dat ze door gebruikers moeten worden goedgekeurd, maar meestal begrijpt de gemiddelde gebruiker

ze niet eens. Het transaktieschema is een dokument van en voor de gebruiker, gesteld in gebruikerstaal. De vertaalslag naar de automatisering moet door de automatiseerders gemaakt worden.
- Het transaktieschema als akseptatiedokument.
De procedure en de dialoog zijn beschreven op het transaktieschema. Dat is dus heel iets anders dan een verslag van een interview, waarvan de gebruiker tijdens de akseptatietest ziet wat de automatiseerders ervan gemaakt hebben. Aan de hand van het transaktieschema kan de gebruiker precies kontroleren of het geleverde klopt met de bestelling.
- Het transaktieschema in een aantal fasen.
Omdat we in de automatisering alleen met de mond belijden dat ontwerpen een iteratief proces is, zijn gebruikers ook gewend aan het gezegde: eens gezegd, blijft gezegd. Dat maakt veel gebruikers schichtig als ze eisen moeten stellen, uitspraken moeten doen of cijfers moeten noemen. Het opstellen van een transaktieschema betekent nog niets. Na dialoogsimulatie ontstaat pas een meer definitieve versie. Soms kan er aan het eind van het technisch ontwerp nog besloten worden een transaktie te bezien. Als het technisch ontwerp is afgerond zijn de transaktieschema's echt definitief.
- De gedetailleerdheid van transaktieschema's.
Op een transaktieschema wordt de procedure aan het beeldscherm vastgelegd. Wanneer Transaktie analyse wordt uitgevoerd in een zeer vroeg stadium, dan zal in het transaktieschema de grote lijn beschreven worden zoals aangegeven in het voorbeeld. Het zal duidelijk zijn dat met een dergelijk transaktieschema nog geen enkele bijdrage is geleverd voor de dokumentatie-set van het logisch ontwerp. Op basis van dit transaktieschema kan een grove Transaktie analyse worden uitgevoerd. Bijvoorbeeld om een indruk te krijgen van het aantal terminaluren per werkplek. Tijdens het verloop van het logisch ontwerp zal op een gegeven moment het konkrete transaktieontwerp in samenwerking met gebruikers plaats vinden. Dan komt de vraag naar voren over de details in het transaktieschema. Laten we eens aannemen dat de gebruiker een zoekargument moet intypen op basis waarvan het systeem de NAW-gegevens van de klant displayd. Laten we verder aannemen dat dit zoekargument nooit cijfers mag bevatten en dat de gebruiker het zeer gemakkelijk kan samenstellen uit enkele letters van de naam en enkele letters van de woonplaats, zie Fig. 42.6. Nu gaat het erom steeds twee zaken te onderscheiden en die vervolgens goed gescheiden te houden.
De eerste zaak is de funktie van het transaktieschema. In het kader van deze paragraaf gaat het om de vastlegging van de procedure in termen van de gebruiker en met een nauwkeurigheid die de gebruiker aanspreekt. Verder heeft het transaktieschema een funk-

tie binnen Transaktie analyse. Binnen Transaktie analyse worden feiten gekwantificeerd met een nauwkeurigheid van laten we zeggen 10%. Het heeft dus niet zoveel zin feiten op een transaktieschema vast te leggen die, kwantitatief bezien, slechts voor enkele procenten van belang zijn. Wanneer iets voor de gebruiker erg wezenlijk is, doch kwantitatief niet interessant, dan dient het toch te worden opgenomen in het transaktieschema. De transaktie-analist kan dan achteraf alsnog besluiten het aspekt niet mee te nemen in de Transaktie analyse.
De tweede zaak die hiervan goed moet worden onderscheiden is de vastlegging van de dialoogstruktuur om op basis daarvan programma's te ontwerpen. Het zal duidelijk, zijn dat bij de vastlegging van de dialoog alle aspekten moeten worden meegenomen. Daarin moeten alle foutsituaties zijn vastgelegd. Het effekt van alle funktietoetsen op alle momenten moet vastliggen. Standaards ten aanzien van schermlay-outs, foutboodschappen, helptoetsen en dergelijke moeten worden meegenomen in het ontwerp. De wijze waarop dit soort zaken wordt vastgelegd is voor de gemiddelde gebruiker ontoegankelijk. Het gaat hier om ontwerpdokumenten. Hij vindt voor hem belangrijke zaken terug in de gebruikershandleiding. De gemiddelde gebruiker heeft echter geen problemen met de standaardplaats van de foutboodschappen of het funktietoetsnummer van de helptoets. Van veel groter belang voor hem, is de tekst van de

Transaktieschema centraal

Transaktienaam: Opzoeken NAW-gegevens

Menselijke handelingen	Transport	Machinale verwerking
Het bepalen en intypen van de verkorte naam.	-----)	Het opzoeken van NAW-gegevens en displayen ervan.
	(-----	
Kontrole op juistheid van de gegevens.		

Fig. 42.6 Voorbeeld van een transaktie.

foutboodschap en de toelichting op het helpscherm. Wanneer de gebruiker, via dialoog-simulatie, konkreet heeft ervaren hoe de procedure aan het beeldscherm verloopt, dan behoeven teksten en toelichtingen niet op een transaktieschema te worden vastgelegd. Kortom, er is nog een leemte op te vullen tussen transaktieschema en ontwerpdokumenten.
Daarmee zijn de grenzen van de gedetailleerdheid van het transaktieschema aangegeven. In het genoemde voorbeeld van het intypen van het zoekargument kan dus op het transaktieschema best alleen de mooi-weersituatie worden weergegeven.
Wanneer de gebruiker de foutsituatie toch in het transaktieschema wil vastleggen, dan wordt het bijvoorbeeld als aangegeven in Fig.42.7.
De informatie-analist moet dus konstant op twee dingen letten: is het van belang voor de gebruiker en/of is het kwantitatief van belang voor transaktie-analyse.
We zullen voor beide mogelijkheden het voorbeeld nog wat verder uitwerken. Stel dat de mogelijkheid bestaat dat er meerdere klanten voldoen aan een zoekargument. Dat probleem zal aan de gebruiker moeten worden voorgelegd. De informatie-analist kan een aantal oplossingen aandragen. Zo kan het zoekargument met een aantal letters worden uitgebreid tot er nog maar één klant aan voldoet. Er kan ook gekozen worden voor de mogelijkheid geldige NAW-gegevens te displayen en de gebruiker de juiste te laten kiezen,

Transaktieschema centraal

Transaktienaam: Opzoeken NAW-gegevens

Menselijke handelingen	Transport	Machinale verwerking
Het bepalen en intypen van de verkorte naam	-----)	Het opzoeken van NAW-gegevens en displayen ervan. Indien geen gegevens gevonden, foutmelding: "Geen klant aanwezig met deze verkorte naam"
Kontrole op juistheid van de gegevens, bij foutmelding korrigeren, of deze bon opzij leggen.	(-----	

Fig. 42.7 Uitbreiding van het voorbeeld.

Transaktieschema centraal

Transaktienaam: Opzoeken NAW-gegevens

Menselijke handelingen	Transport	Machinale verwerking
Het bepalen en intypen van de verkorte naam	-----)	Het opzoeken van NAW-gegevens en displayen ervan. Indien meerdere voldoen, keuze-scherm met lijst van NAW-gegevens.
	(-----	
Bij keuzescherm regel-nummer intypen	-----)	Het opzoeken van NAW-gegevens en displayen ervan.
	(-----	
Kontrole op juistheid van de gegevens.		

Fig. 42.8 Verdere uitbreiding van het voorbeeld.

waarna de mooi-weersituatie weer doorloopt, zie Fig. 42.8. Voorlopig kan een alternatief gekozen worden en worden opgenomen in het transaktieschema.
Het kan zijn dat het werken met verkorte namen en bjbehorende keuze-schermen bij de gebruikers reeds bekend is en bovendien de afhandeling door het systeem reeds standaard is binnen het bedrijf of de afdeling. Dan zou dat aspekt voor deze gebruikers best weggelaten mogen worden op het transaktieschema. Het zal echter duidelijk zijn dat het kwantitatief van belang is voor Transaktie analyse. Immers, het systeem moet een lijst samenstellen, meestal gesorteerd op alfabet, het keuzescherm ophalen en displayen. Wanneer de kans bestaat dat de lijst niet op een scherm past, moeten er nog vervolgschermen gedisplayed kunnen worden. Zou dit alles zich enkele keren per dag voordoen dan zal het kwantitatief nog niet interessant zijn. Wanneer het voorkomt bij 10% van de transakties dan is het zeker de moeite waard om het vast te leggen op het transaktieschema.
Het zal duidelijk zijn dat op deze manier transaktieschema's snel onleesbare dokumenten kunnen worden, als ze al geschreven kunnen worden. Op dat aspekt komen we nog uitgebreid terug bij "Transaktieschema's in komplexe situaties".
Wanneer het gaat om de mate van gedetailleerdheid, zijn er geen algemene regels te geven. Het kan zijn dat het qua afspraken met

Transaktieschema centraal		
Transaktienaam: Opzoeken van NAW-gegevens		
Menselijke handelingen	Transport	Machinale verwerking
Het bepalen en intypen van de verkorte naam.	-----)	Het opzoeken van NAW-gegevens en het displayen ervan.
	(-----	
Kontrole van de gedisplayde gegevens. Het intypen van de eerste orderregel.	-----)	
Het overige deel van de transaktie.		
Aan het einde van het transaktieschema:		
		Indien er meerdere NAW-gegevens voldoen, een keuzescherm met een lijst van NAW-gegevens displayen.
	(-----	
Regelnummer van de juiste klant bepalen en intypen.	-----)	

Fig. 42.9 Uitsprongen op een transaktieschema.

de gebruikers en qua uitvoering van Transaktie analyse verantwoord is om alleen de mooi-weersituatie vast te leggen. In dat geval moet later nog het een en ander worden gedaan aan de exakte vastlegging van de dialoogstruktuur. Per bedrijf kan die wijze van vastleggen verschillen. Meestal zal die vastlegging aansluiten bij gekozen dokumentatie-methoden. Hoe meer er wordt vastgelegd op het transaktieschema, hoe onoverzichtelijker het voor de gebruiker wordt. Wanneer te weinig wordt vastgelegd bestaat de kans dat de resultaten van Transaktie analyse aan waarde verliezen.

Zoals in het begin van deze paragraaf reeds is aangegeven, bestaat er een funktioneel verschil tussen het transaktieschema en dokumenten voor het programma ontwerp. Kort samengevat kunnen we vaststellen dat bij het opstellen van transaktieschema's gelet moet worden op de waarde van het dokument voor de gebruiker en voor Transaktie analyse. Het opzetten van de ontwerpspecifikaties voor de dialoog gebeurt later, maar wel op basis van de transaktieschema's.

- Transaktieschema's in komplexe situaties.

In de vorige paragraaf is reeds aangegeven dat transaktieschema's onleesbaar kunnen worden door teveel details. Transaktieschema's kunnen ook onleesbaar worden omdat de transakties ingewikkeld in elkaar zitten. Er zijn een aantal mogelijkheden om ook in die situaties de zaak overzichtelijk te houden. We onderscheiden daartoe enkele soorten transakties.

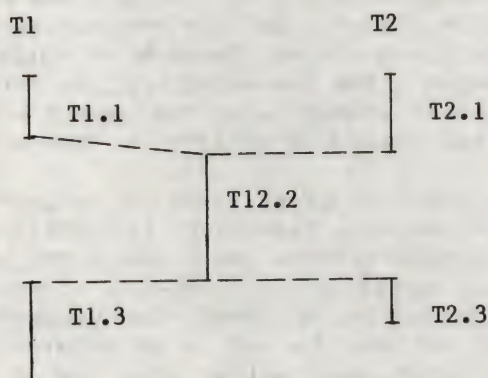
Rechtlijnige transakties.

Dit zijn transakties die altijd op dezelfde manier verlopen, zonder uitsprongen of tussenvoegingen. Het transaktieschema beschrijft eenvoudig de op elkaar volgende handelingen. Zoals in de vorige paragraaf is aangegeven hoeft dat niet te betekenen dat er geen uitsprongen zijn. Foutsituaties kunnen in alle transakties optreden en moeten opgevangen worden. De informatie-analist stelt vast of die situaties kwantitatief relevant zijn voor Transaktie analyse of niet. Deze transaktieschema's worden door de transaktie-analist omgezet in detailschema's en leveren per transaktie resultaten op.

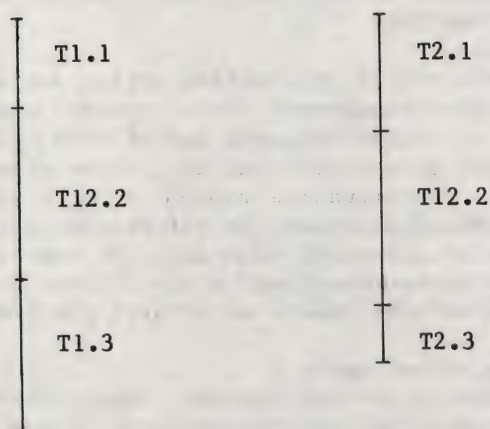
Transakties met enkele uitsprongen.

We nemen weer het voorbeeld van het opzoeken van de NAW-gegevens, zie Fig. 42.9. De situatie met het keuzescherm is een voorbeeld van een uitsprong. De resultaten van Transaktie analyse blijven hetzelfde wanneer men de uitsprong aan het eind van het transaktieschema opneemt.

Het aantal uitsprongen dat aan het einde van het transaktieschema wordt opgenomen wordt alleen beperkt door de gewenste overzichtelijkheid van het geheel. In situaties met komplexe uitsprongen en



Struktuur van twee transakties.



Oplossing met twee transakties.

Fig. 42.10 Twee transakties met een gemeenschappelijk deel.

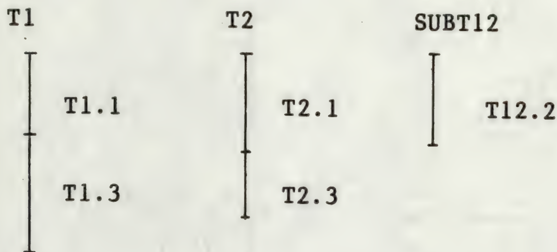


Fig. 42.11 Oplossing met twee transakties en een subtransaktie.

uitsprongen binnen uitsprongen kan men op dezelfde wijze blijven werken maar bij uitgebreide uitsprongen is het beter er subtransakties van te maken. Een subtransaktie is een deel van een transaktie waarvoor een apart detailschema wordt gemaakt en het rekenproces wordt uitgevoerd. De resultaten van de oorspronkelijke transaktie en van een of meer subtransakties worden door de transaktie-analist weer samengevoegd in een combitransaktie. Een opslitsing in subtransakties is ook nuttig wanneer verschillende transakties bepaalde gedeelten gemeenschappelijk hebben. Wanneer men dat gemeenschappelijk deel slechts eenmaal wil vastleggen, kan dat in de vorm van een subtransaktie. De transaktie-analist voegt later de resultaten toe aan de desbetreffende transakties. In het volgende voorbeeld gaat het om twee transakties T1 en T2. Zie Fig. 42.10. Het middelste deel is voor beide hetzelfde, alleen de aan- en uitloop zijn verschillend. De informatie-analist kan twee transaktieschema's maken. Eén voor T1, bestaande uit de delen T1.1, T12.2 en T1.3 en één voor T2 bestaande uit T2.1, T12.2 en T2.3. De transaktie-analist maakt twee detailschema's en voert voor beide transakties het rekenproces uit. Zo ontstaan per transaktie de standaard resultaten. De informatie-analist kan ook besluiten van het gemeenschappelijke deel een subtransaktie te maken. Hij maakt dan drie transaktieschema's zoals in Fig. 42.11 is aangegeven. In de kop van het transaktieschema van de sub-

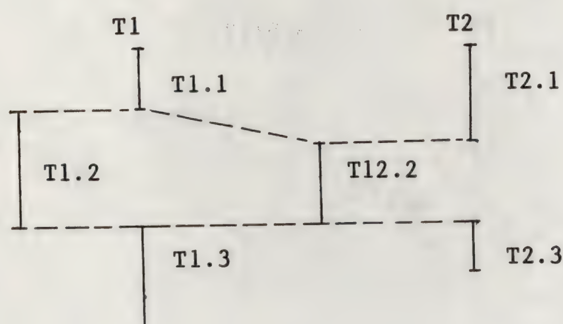


Fig. 42.12 Struktuur van de twee transakties

transaktie wordt aangegeven van welke transakties de subtransaktie deel uitmaakt. De transaktie-analist maakt drie detailschema's en voert drie keer het rekenproces uit. Vervolgens worden de resultaten van T1 en SUBT12.2 samengevoegd in een combitransaktie COMT1 en die van T2 en SUBT12.2 in COMT2. Uiteraard is de naamgeving van de transakties volkomen vrij en kan die worden aangepast aan de bestaande standaards.
Laten we het voorbeeld nog wat ingewikkelder maken door aan te nemen dat in T1 een keuze gemaakt kan worden tussen deel T1.2 en deel T12.2. Zie Fig. 42.12.
De informatie-analist kan T1 op een transaktieschema weergeven zoals in het vorige voorbeeld. Op dat schema zal dus ergens zoiets staan als: "in sommige gevallen" of "soms", en dan volgt het deel T1.2 en vervolgens zoiets als: "in alle andere gevallen" en dan volgt het deel T12.2. Wanneer op dat moment al iets bekend is over aantallen is het natuurlijk beter om op het transaktieschema meteen percentages te noemen. Bijvoorbeeld: "in 30% van de gevallen" of "in 70% van de gevallen".
De informatie-analist kan ook kiezen voor de oplossing met de subtransaktie T12.2. Op het transaktieschema van T1 kan dan bijvoorbeeld staan "in 30% van de gevallen" en dan volgt T1.2 en "in 70% van de gevallen SUBT12.2". Een verwijzing naar een subtransaktie dus. De informatie-analist kan er ook voor kiezen om van T1.2 een subtransaktie te maken.

De keuze tussen de diverse oplossingen wordt bepaald door het inzicht van de analist. Hij overweegt daarbij de komplexiteit van de transakties en eventuele subtransakties, de overzichtelijkheid voor de gebruiker, de manier waarop de gebruiker aankijkt tegen transaktiedelen, de dokumentatiestandaards van de ontwerpafdeling. Het is van belang om goed vast te stellen hoe de gebruiker de diverse transakties ziet. Het komt regelmatig voor dat voor de ontwerpers twee transakties praktisch hetzelfde zijn omdat de programma's die erbij horen elkaar voor 95% overlappen. De gebruiker ziet het als twee totaal verschillende transakties omdat ze in totaal verschillende omstandigheden worden uitgevoerd. Dat kan een argument zijn om toch maar twee volledige transaktieschema's te maken. Met behulp van de kopieermachine of de tekstverwerker is dat eenvoudig te regelen. Het zal duidelijk zijn dat bij gebruikmaking van subtransakties de kwantiteiten over de subtransakties moeten worden doorgegeven aan de transaktie-analist. Dat kan eenvoudig door in de kop van een transaktieschema van een subtransaktie aan te geven van welke transakties de subtransaktie deel uitmaakt en in hoeveel procent van de gevallen. Soms moeten transaktieschema's gemaakt worden terwijl de dialoogstruktuur al helemaal vastligt. De komplexiteit van sommige dialoogstruktuurdiagrammen lijkt een sequentieel transaktieschema onmogelijk te maken. In de praktijk blijkt dit altijd te gaan, al zou men per scherm een transaktieschemablad moeten maken. De transaktie-analist wil dan alleen per scherm nog weten wat de kans is dat het wordt gebruikt in een transaktie en hoe vaak het doorlopen wordt. Voor hem is dan ieder transaktieschemablad een deel van het detailschem met een bepaalde kansfaktor. Voor het rekenprogramma maakt de volgorde op het detailschema niets uit.
Wellicht ten overvloede zij er nogmaals op gewezen dat steeds in het oog moet worden gehouden in hoeverre alle onderdelen kwantitatief interessant zijn. Zie het punt "De gedetailleerdheid van transaktieschema's".

- Decentrale transaktieschema's

In het transaktieschema wordt in de rechterkolom de verwerking door de computer beschreven. In decentrale situaties kan het zijn dat de verwerking moet worden uitgevoerd op twee computers, die via een netwerk gekoppeld zijn. Het is mogelijk het transaktieschema naar rechts uit te breiden met twee kolommen. Dan ontstaat het decentrale transaktieschema. De informatie-analist kan dan aangeven welke verwerking decentraal en welke centraal plaatsvindt, maar het is de vraag of dat nodig is. Het transaktieschema is in de eerste plaats gebruikersdokument en dus niet bedoeld om de technische realisering van de verwerking precies weer te geven. Het feit dat de verwerking op twee gekoppelde computers plaatsvindt zou transparant moeten zijn voor de gebruiker. In de

Transaktieschema centraal		
Transaktienaam: Bezoekersregistratie		
Menselijke handelingen	Transport	Machinale verwerking
Vragen om legitimatie Lezen en intypen zoek-gegevens	----)	Opzoeken persoonsgegevens. Indien gevonden, displayen, anders aanmeldscherm displayen.
	(----	
Lezen persoonsgegevens of intypen persoons-gegevens	----)	Beginscherm displayen of gegevens verwerken en dan beginscherm displayen.
	(----	

Fig. 42.13 Centraal transaktieschema BEZREG.

praktijk is het echter meestal zo dat we zo'n verwerking niet transparant kunnen houden tengevolge van de lange responsetijden. Als de informatie-analist dat voorziet, kan hij natuurlijk best de gescheiden verwerking vanaf het begin aan de gebruiker uitleggen en vastleggen op het transaktieschema. In Fig. 42.13 is een kort transaktieschema weergegeven, waarvan Fig. 42.14 de decentrale versie is.
Als vanaf het begin bekend is, dat het gaat om decentrale verwerking kan de informatie-analist zelf kiezen uit de twee mogelijkheden. Als de hele systeemopzet nog ontworpen moet worden is er niet veel keus. Dan zal de informatie-analist wel een centraal transaktieschema moeten maken. In dat geval levert hij een centraal transaktieschema af aan de transaktie-analist. Tijdens het technisch ontwerp zorgt de transaktie-analist dan voor de korrekte detailschema's. Of er, nadat het definitieve systeemontwerp gereed is, alsnog decentrale transaktieschema's moeten worden gemaakt, is ter beoordeling van de informatie-analist. Als het voor de gebruiker niet nodig is, voor de transaktie-analist zeker niet.

42.4 Transaktie-ontwerp in verschillende omgevingen

In paragraaf 41.5 is een aantal omgevingen gedefinieerd. Uiter-

Transaktieschema decentraal

Transaktienaam: Bezoekersregistratie

Menselijke handelingen	Transport	Machinale verwerking	Transport	Machinale verwerking
Vragen om legitimatie Lezen + inty- pen zoekgeg.	----)			
		Opzoeken, indien niet gevonden, opvragen op hoofdkantoor	----)	
				Opzoeken pers. geg. Retour: gegevens of melding: geen gegevens aanwezig
			(----	
		Displayen van gegevens of van aanmeldscherm		
Lezen pers.geg.	(----			
of intypen per- soonsgegevens	----)			
		Beginscherm displayen of geg. verwer- ken, centraal en lokaal, en dan begin- scherm dis- playen.		
	(----		----)	
				Nieuwe pers. gegevens opne- men in bestand

Fig. 42.14 Decentraal transaktieschema BEZREG.

aard werd de basis voor het aanbrengen van de onderscheidingen gevormd door de methoden die we behandelen. We gaan nu de werkwijze in de diverse omgevingen verder uitwerken aan de hand van Fig 42.17 tot 42.25.
Ieder figuur bestaat uit drie delen. Het linker deel geeft de gebruikerswereld weer met begrippen als afdelingen, know-how, dossiers, gegevens. Het rechterdeel duidt de automatiseringsafdeling aan in termen van know-how, ontwerpresultaten, data dictionary, bestanden.
Tussen deze beide delen bevindt zich het kommunikatiegebied. Daar vinden we aktiviteiten en dokumenten waar gebruikers en de automatiseerders samen mee bezig zijn.
De zwarte enkele pijlen geven aan:
- hoe dokumenten of gegevens ontstaan uit aktiviteiten,
- hoe dokumenten of gegevens leiden tot aktiviteiten,
- hoe dokumenten worden opgeborgen.

De zwarte dubbele pijlen geven een vergelijking aan. Een L duidt op een logische vergelijking, dus een kontrole op kompleetheid, afwijkingen van het reeds aanwezige ontwerp, funktionele verschillen etc. Een P slaat op een performance-vergelijking. Daarbij gaat het om overduidelijke afwijkingen van de gestelde eisen, zoals is aangegeven in de paragraaf over responsetijden. De witte pijlen geven know-how aan. Het gaat dan om de materiedeskundigheid van de gebruiker of om automatiseringsvakkennis. Bij iedere figuur kan worden vastgesteld dat transaktieontwerp en Transaktie analyse geen bestaande methoden, aktiviteiten of dokumenten vervangen. Beide methoden dienen om gebruikersinbreng te realiseren. De pijlen geven aan hoe die inbreng de ontwerpen van de automatiseerders kan beinvloeden en hoe de gebruikers konkrete informatie krijgen over de toekomstige situatie.
Het gaat om transaktie-ontwerp, dat niet uitloopt op een netwerkontwerp. Dat betekent dat het gaat om een mainframe of een mini met lokale terminals of met remote aangesloten terminals waarvan de lijnen al aanwezig zijn. De ontwerpprocedure richt zich dus op de ergonomische aspekten en de technische aspekten zoals performance, systeembelasting, maar niet op de hoeveelheid verkeer en de lokatie van de gegevens. De netwerkontwerpaspekten sluiten trouwens wel aan op de te beschrijven procedures. Het gaat nu echter om omgevingen waar geen netwerk ontworpen hoeft te worden. Eerst worden in Fig. 42.15, Transaktie-ontwerp in beeld, de onderdelen van het transaktie-ontwerp aangegeven. In de volgende figuren wordt behandeld hoe die delen moeten worden gerelateerd aan de andere ontwerpaktiviteiten. In Fig. 42.17 tot 42.21 gaat het om een M1/M2-omgeving, waar niet gewerkt wordt met databases en een data dictionary. In Fig. 42.22 tot 42.25 gaat het om de M3-omgeving waar allerlei ontwerpmethoden en -gereedschappen be-

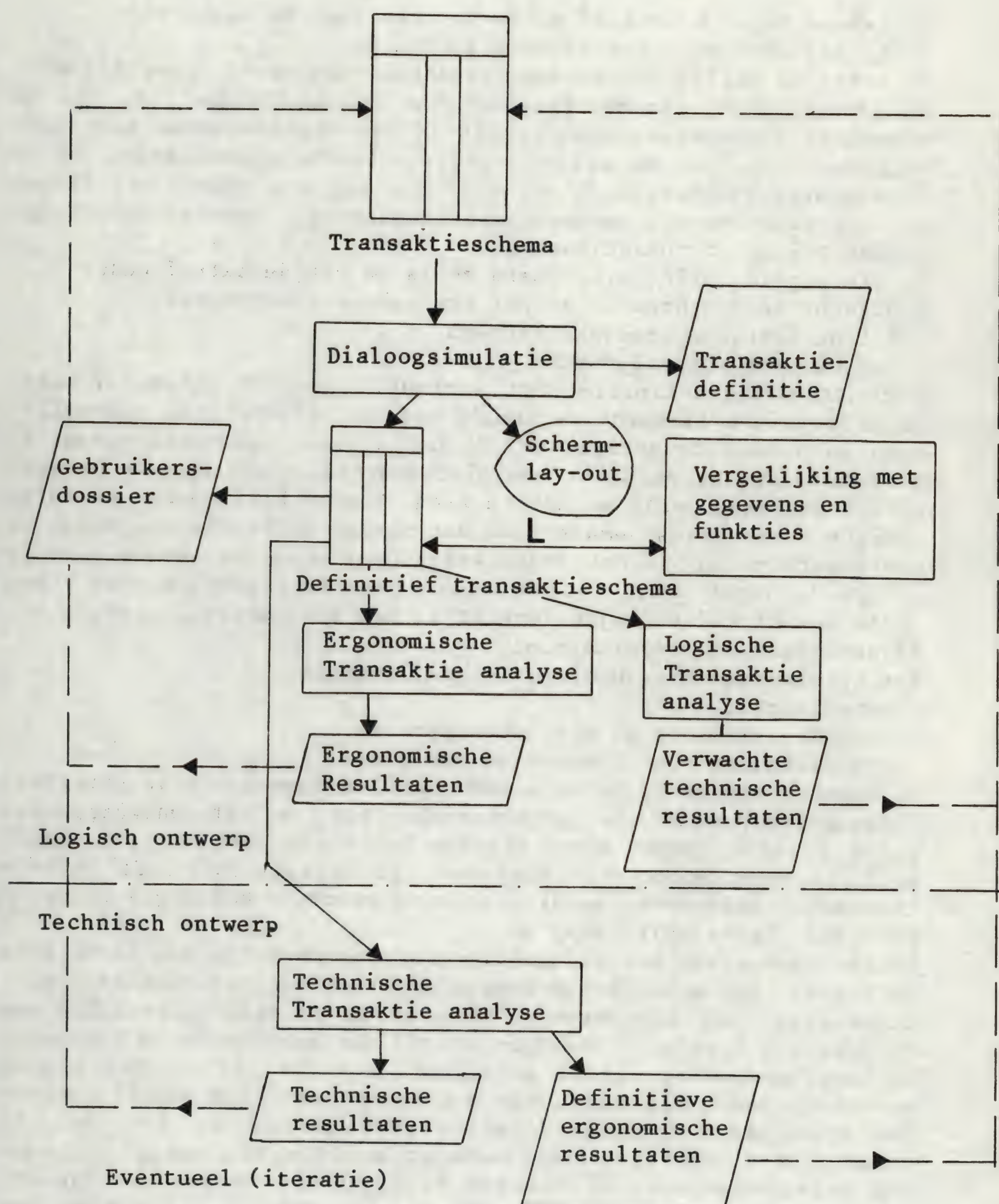


Fig. 42.15 Transaktie-ontwerp in beeld

schikbaar zijn. We zullen nu de diverse figuren toelichten.
- Fig. 42.15 Transaktie-ontwerp in beeld.
De start is altijd het transaktieschema. Het wordt opgesteld door de gebruiker in samenwerking met de informatie-analist. Zie de paragraaf Transaktieschema's. Dit is het startdokument voor dialoogsimulatie. In de meeste systeemontwikkelingsmethoden en in alle praktijksituaties is de schermlay-out het eerste wat de gebruiker ziet van een interaktieve toepassing. Transaktie-ontwerp begint bij het transaktieschema.
Na dialoogsimulatie, uitgevoerd zoals in het desbetreffende hoofdstuk is beschreven, is per transaktie beschikbaar:
- het definitieve transaktieschema
- een of meer schermlay-outs
- de transaktiedefinitie, het verband tussen de schermlay-outs.
De definitieve transaktieschema's met de bijbehorende schermlay-outs gaan naar de gebruikers. De definitieve transaktieschema's, de schermlayouts en de transaktiedefinities gaan naar de ontwerpers. Daar zou, zoals we zullen zien, kunnen blijken dat de definitieve transaktieschema's toch wat minder definitief zijn dan de gebruikers dachten. Het transaktieschema plus de schermlay-outs vormen de input voor Transaktie analyse. De schermlay-outs komen op de een of andere wijze terecht in een schermbibliotheek, zie de paragraaf Dialoogsimulator.
Het transaktieschema heeft drie hoofdfunkties:
- gebruikersdokument
- gebruikersinbreng in het ontwerpproces
- startdokument voor Transaktie analyse.
De funktie van gebruikersdokument wordt behandeld in de paragraaf Transaktieschema's. De gebruikersinbreng in het ontwerpproces wordt in de volgende serie figuren behandeld. Dat het startdokument is voor Transaktie analyse ligt opgesloten in de methode Transaktie analyse en wordt uiteraard eveneens behandeld in de paragraaf Transaktieschema's.
De levensduur van het transaktieschema is verschillend in de drie situaties. Als gebruikersdokument ontstaat het transaktieschema tijdens het logisch ontwerp. Wanneer na de dialoogsimulatie een definitieve versie is ontstaan, wordt het opgeborgen in het automatiseringsdossier van de gebruiker. Daar blijft het tot aan de overdrachtfase. Dan wordt het opgeleverde systeem vergeleken met het transaktieschema. Tijdens het overige deel van het logische ontwerp of tijdens het technische ontwerp kan het nodig zijn alsnog wijzigingen aan te brengen bijvoorbeeld tengevolge van onhaalbare responsetijden of technische beperkingen die tijdens het logische ontwerp niet bekend waren.
Als dokument voor de gebruikersinbreng in het ontwerpproces funktioneert het alleen tijdens het logisch ontwerp. De op het trans-

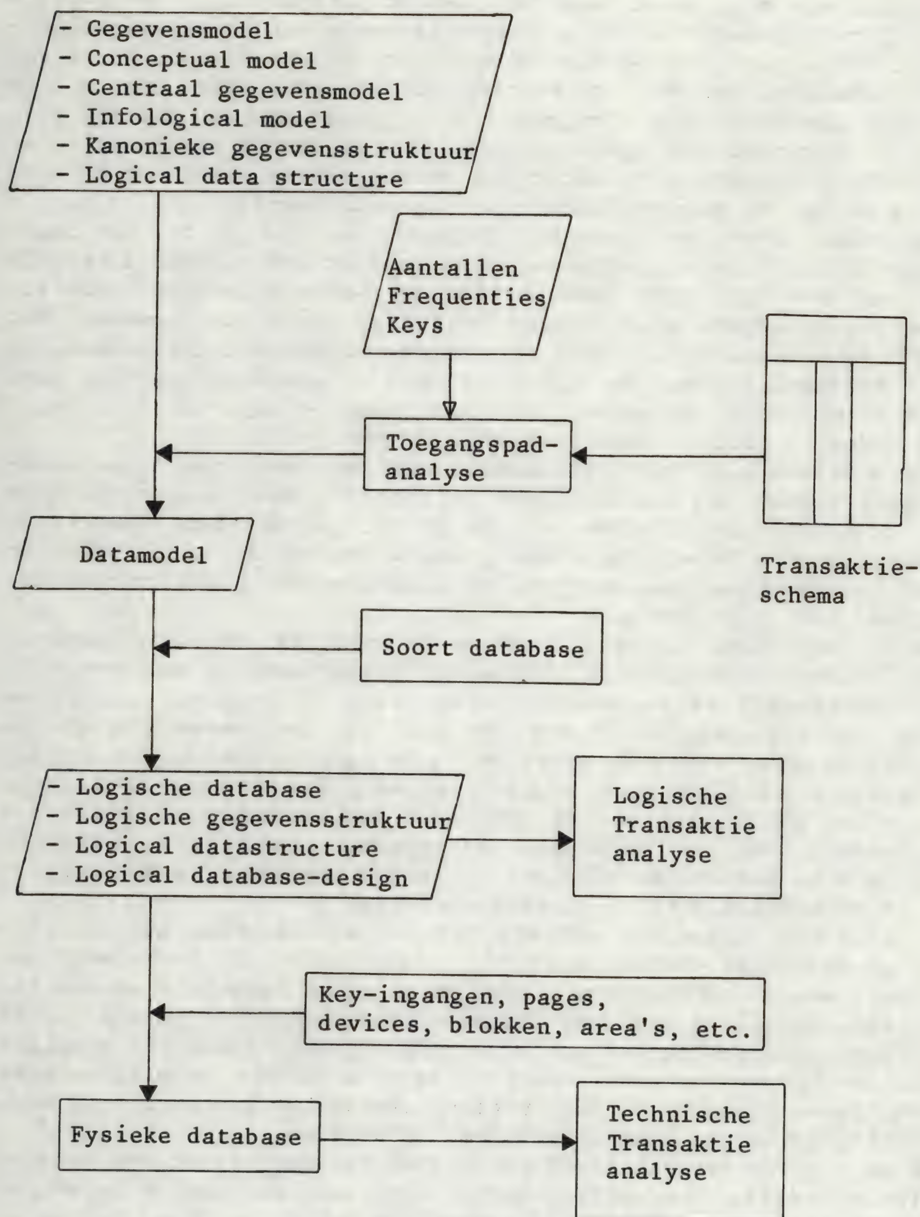
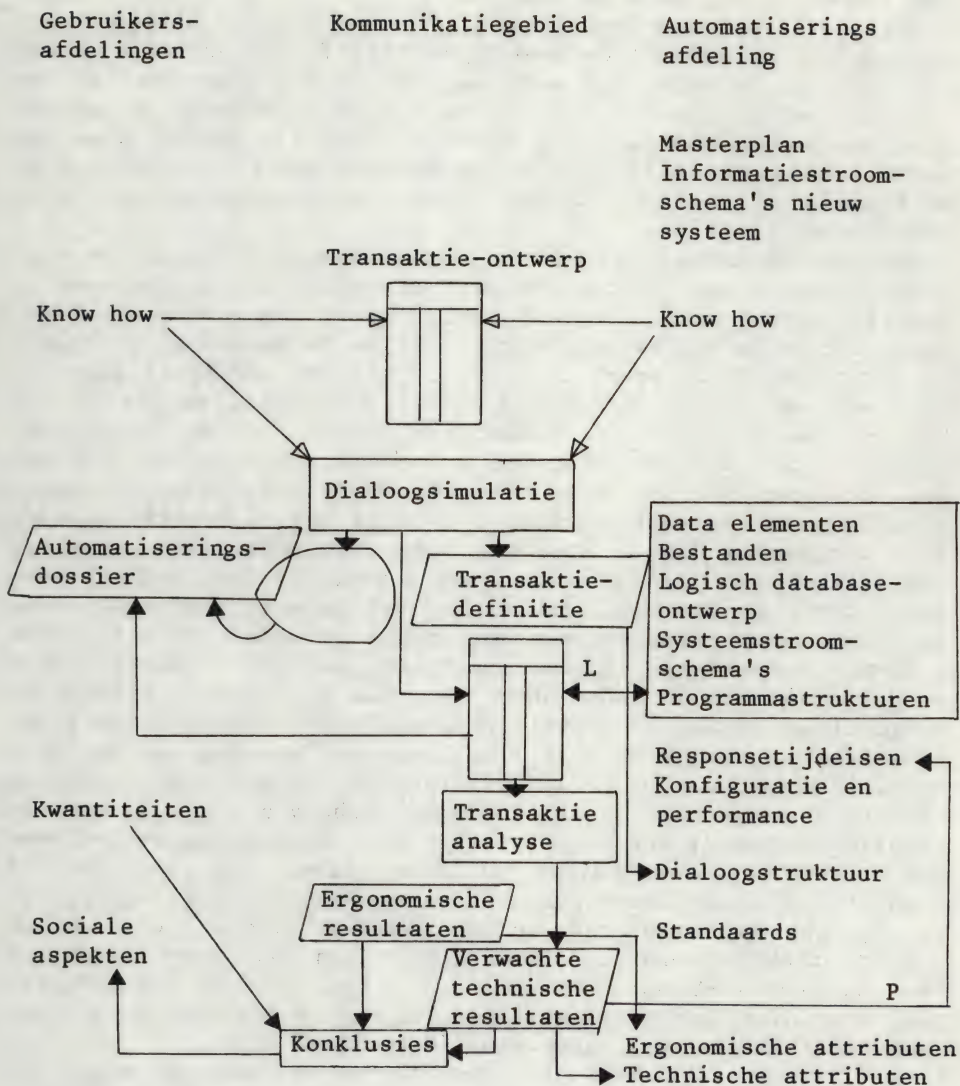


Fig. 42.16 Transaktie- en database-ontwerp.

aktieschema aangegeven verwerking wordt vergeleken met de ontworpen funktiemodellen en de genoemde gegevens met de gegevensmodellen. Bij verschillen worden de modellen aangepast of er wordt van het transaktieschema, in overleg met de gebruiker, een nieuwe versie gemaakt. Bij ingrijpende wijzigingen kan het nuttig zijn de dialoogsimulatie opnieuw uit te voeren. Na de vergelijking en eventuele aanpassing verloopt het ontwerpproces verder op de normale wijze. De gebruikersinbreng is gerealiseerd.
Daarnaast levert het transaktieschema een bijdrage tot het database-ontwerp. De toegangspadanalyse behoort ook gebaseerd te zijn op het dialoogontwerp. Soms worden databases volgens de fraaiste regels ontworpen, maar zonder te letten op de performance. Het technische ontwerp kan niet alle ontwerpkronkels rechttrekken. Op het transaktieschema is aangegeven hoe de gegevens gebruikt worden en met welke verwachte responsetijden. In Fig. 42.16 is een en ander in grote lijnen in kaart gebracht.
Als startdokument voor Transaktie analyse leeft het transaktieschema totdat het detailschema is gemaakt, maar meestal blijven de transaktieschema's bewaard in het projektdossier. Wanneer de tekst op het detailschema een goede weergave is van de tekst op het transaktieschema, heeft de transaktie-analist het transaktieschema niet meer nodig.
Het detailschema kan op diverse manieren worden ingevuld. Wanneer de Transaktie analyse alleen wordt uitgevoerd om de terminal-transaktietijd en de samenstelling ervan te bepalen, spreken we van een ergonomische Transaktie analyse. Het verwerkingsproces wordt niet in rekening gebracht, gewenste responsetijden worden ingesteld. Tijdens het logisch ontwerp kunnen ook technische aspekten al een rol spelen. Het gaat dan om zaken als verwachte responsetijden en verwachte systeembelasting. Deze verwachte technische resultaten zijn per definitie gebaseerd op het logische gegevensontwerp of database-ontwerp.
Tijdens het technische ontwerp kan het detailschema uitgebreider en nauwkeuriger worden ingevuld. Nu kunnen ook de technische aspekten van de beeldschermlay-out en de soort terminals worden ingevuld. Op basis van het fysieke gegevensontwerp kunnen nu de verwerkingsaspekten definitief worden begroot, eventueel kompleet met de systeemoverhead. Zowel de technische als de ergonomische resultaten bereiken hun definitieve waarden en kunnen alsnog aanleiding geven het transaktieschema te herzien.
In de figuren van Fig 42.17 tot Fig 42.21 gaat het om een eenvoudige situatie, waar niet gewerkt wordt met een data-dictionary/directory, systeemontwikkelingsmethoden of grote, complexe databases. Of, om het in de termen van de paragraaf Methoden en omgevingen te zeggen: een M1/M2-Cx-omgeving. Qua hardware zal het meestal gaan om een mainframe voor eenvoudige maar soms massale



Omgeving: Centraal systeem of stand alone decentraal systeem.
Afgebakend projekt, geen netwerkontwerp.
Logische Transaktie analyse.

Fig. 42.17 Transaktie ontwerp tijdens het logisch ontwerp

toepassingen of om een minicomputer. Beide kunnen voorzien zijn van een eenvoudig database systeem.
- Fig. 42.17 Transaktie-ontwerp tijdens het logisch ontwerp.
Het logisch ontwerp is de meest geschikte fase voor het transaktie-ontwerp. In deze fase ontstaan ook de logische funkties van het informatiesysteem en het logische gegevensontwerp. Op dat moment is een goede afstemming mogelijk. Voor het korrekt funktioneren van het definitieve transaktieschema maakt het niet uit op welk moment tijdens het logisch ontwerp het transaktie-ontwerp plaatsvindt.
Wanneer alle transaktieschema's beschikbaar zijn, kunnen de vermelde gegevens en funkties vergeleken worden met de gegevens en funkties die ontworpen zijn op basis van de uitgevoerde analyses. Aangezien de analyses zijn uitgevoerd om de bestaande processen en aktiviteiten in kaart te brengen, zijn de ontworpen gegevens en funkties meestal gebaseerd op de bestaande, handmatige of batch-situatie. Kreatieve gebruikers zouden bij de informatieanalisten best aan kunnen komen met nieuwe aktiviteiten, die mogelijk worden door het werken met beeldschermen. Zodoende zouden er op diverse transaktieschema's meer of andere funkties kunnen zijn aangegeven dan de analyses deden vermoeden. Het uitvoeren van het transaktie-ontwerp zo vroeg mogelijk in het logisch ontwerp, heeft als voordeel dat er bij het ontwerp van funkties en gegevens direkt rekening mee kan worden gehouden. Anders moeten achteraf aanpassingen worden ingevoerd, hetgeen de konsistentie en de logika van de strukturen niet ten goede komt. Tijdens de vergelijking tussen de transaktieschema's en de funktie- en gegevensontwerpen komen meestal enige verschillen naar voren. Deze verschillen dienen te worden beoordeeld. Uiteindelijk worden de ontwerpen aangepast of de transaktieschema's. Wijzigingen in transaktieschema's kunnen natuurlijk niet worden aangebracht zonder overleg met de gebruiker. Hij heeft immers een kopie in zijn dossier als akseptatiedokument. Het aanbrengen van ingrijpende wijzigingen kan betekenen dat de dialoogsimulatie voor een of meer transakties opnieuw wordt gedaan. In dit soort situaties komt het voordeel van dialoogsimulatie ten opzichte van prototyping weer naar voren: het kost per transaktie erg weinig tijd om de gebruikers de gewijzigde situatie te laten zien.
Niet in alle systeemontwikkelingsmethoden ontstaan gegevens- en programmastrukturen tijdens het logisch ontwerp, met een voldoende graad van gedetailleerdheid om de genoemde vergelijking mogelijk te maken. In die gevallen wordt de vergelijking uitgesteld tot het technisch ontwerp, evenals alle andere nog te noemen aktiviteiten. Het gaat erom de logische, niet aan een machine gebonden ontwerpen te toetsen, onafhankelijk van de naam van de fase waarin ze ontstaan.

Een volgend punt in de figuur is de responsetijdeisen die uit dialoogsimulatie naar voren zijn gekomen. Niemand kan tijdens het logisch ontwerp voorspellen of de responsetijden een of twee seconden zullen worden. Met andere woorden, elke schatting zou er gemakkelijk 100% naast kunnen zitten. In de praktijk blijkt echter dat er regelmatig systemen gebouwd zijn, waarvan sommige responsetijden 1000 tot 10.000% te lang zijn. Te lang slaat dan op de verwachting van de eindgebruikers. In het hoofdstuk Dialoogsimulatie wordt behandeld hoe responsetijdeisen tot stand komen. Achteraf blijkt altijd dat extreem lange responsetijden te voorzien waren geweest tijdens het logisch ontwerp. Een gesorteerde deelverzameling uit 10.000 entiteiten staat niet binnen een seconde op het scherm. Tijdens het logisch ontwerp moeten dus de responsetijdeisen uit dialoogsimulatie gelegd worden naast de struktuur van het logisch gegevensontwerp. Situaties die in de verste verte niet haalbaar zullen zijn, worden nu besproken met de gebruikers. In de dialoogsimulator worden voor die situaties de realistisch geachte responsetijden ingesteld. Hoe lang 20 sekonden duren wordt dan pas goed voelbaar! Vervolgens mag de gebruiker kiezen uit de volgende mogelijkheden:

- toch op deze wijze de transaktie laten realiseren,
- een alternatieve transaktie ontwerpen met bijvoorbeeld andere zoekargumenten of deelverzamelingen,
- afzien van deze transaktie.

Onafhankelijk van de keuze is nu in ieder geval het probleem van de irritatie achteraf voorkomen. Nogmaals, het gaat om te lange responsetijden ten gevolge van het logisch ontwerp. Voor responsetijdeisen die haalbaar lijken, volgt nog een kontrole tijdens het technisch ontwerp.

In welke mate de kontrole op responsetijdeisen tijdens het logisch ontwerp zinvol is, hangt voor een belangrijk deel af van de ervaring van de systeemontwerpers. Bekende deskundigen achten het heel goed mogelijk. In (7) lezen we op pagina 366 dat, nadat in het logisch ontwerp, de toegangspaden tot de gegevens in kaart zijn gebracht op Transaction Usage Maps, "An "empirical" estimate of possible physical response time van also be made, if a preliminary physical file design or physical data base design is considered. This response time may be evaluated in terms of the end user's absolute response time (performance) requirement". Op dat moment is er nog geen sprake geweest van dialoogontwerp, soort terminals of netwerkontwerp. Daar wordt dus tijdens het logisch ontwerp al gesproken over responsetijden, zonder dat de gebruiker konkrete eisen heeft kunnen stellen via zoiets als dialoogsimulatie en de automatiseerders alleen nog maar beschikken over een gegevensmodel.

Vervolgens komen we bij de konfiguratie en de performance. Uit de

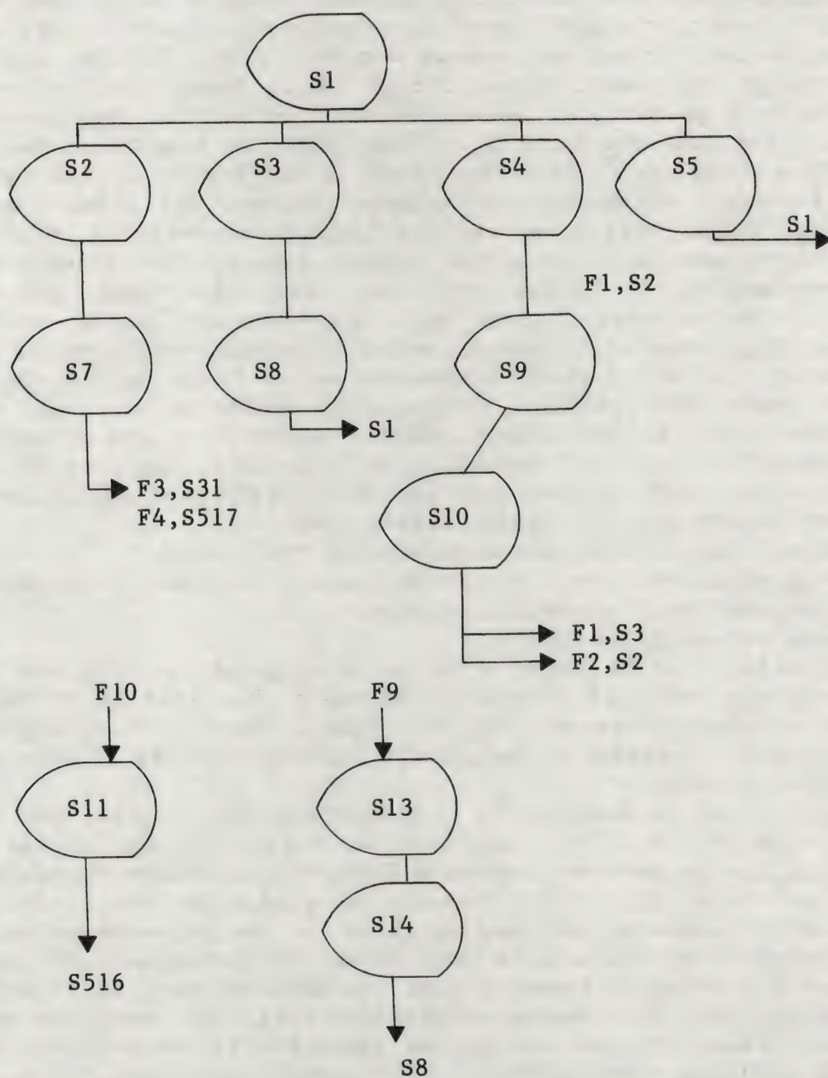
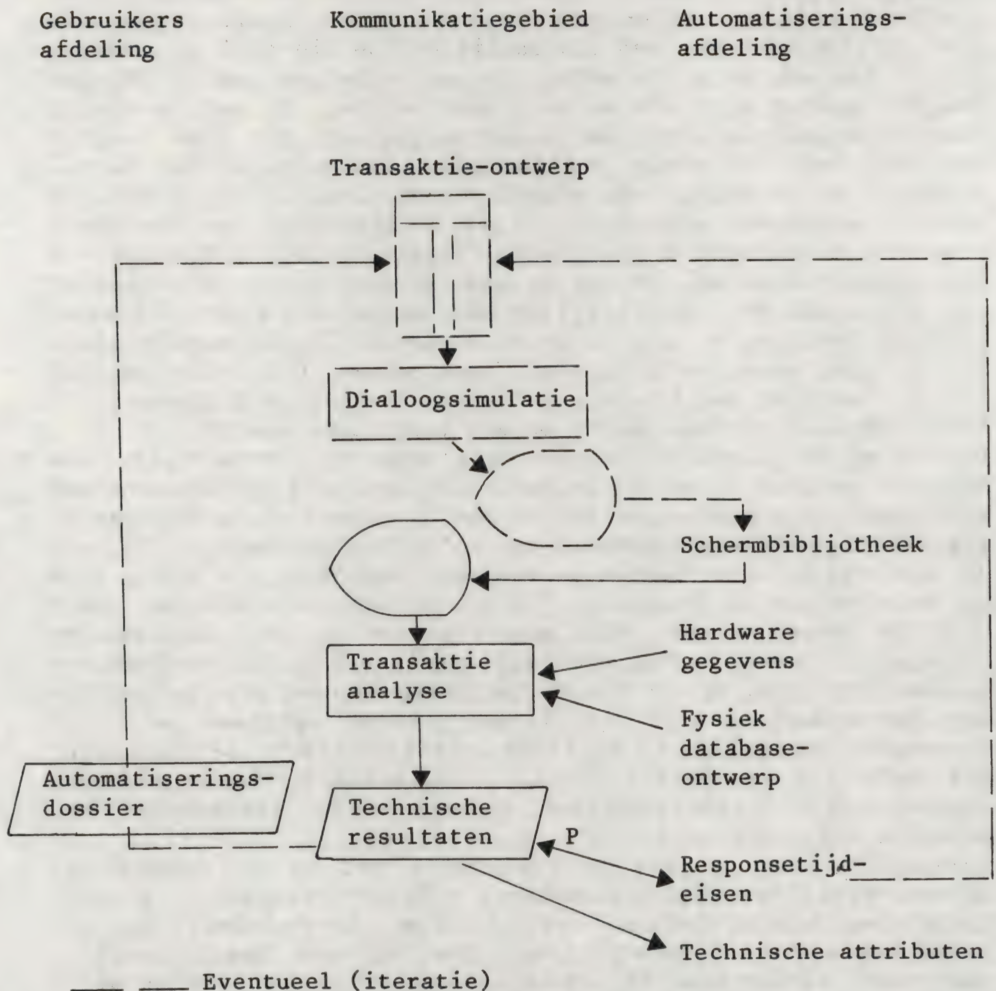


Fig. 42.18 Voorbeeld van een dialoogstruktuur.

logische Transaktie analyse komen ergonomische en verwachte technische resultaten. De ergonomische resultaten bevatten gegevens over de tijdsduur van een transaktie en hoe die tijd is samengesteld. Daaruit is te berekenen hoeveel uren per dag de eindgebruiker achter zijn beeldscherm moet zitten. In het hoofdstuk Transaktie analyse wordt dit verder uitgewerkt. Uit deze resultaten valt ook het aantal beeldschermen en eventueel het aantal printers te berekenen. De technische resultaten van Transaktie analyse geven aan hoeveel I/O's per tijdseenheid een transaktie vraagt. Gekombineerd met het aantal terminals levert dit konkrete performance-eisen op. Of het nu gaat om een grote, een mini- of een microcomputer, steeds blijkt het aantal I/O's per tijdseenheid het knelpunt te zijn in de performance. Tegelijkertijd blijken er geen eisen te zijn, omdat men zonder Transaktie analyse niet verder komt dan termen als "voorraadbeheer op een paar beeldschermen" of "data entry op een multi-user micro".
Bij alle interaktieve toepassingen moet de opeenvolging van schermen precies zijn vastgelegd. Die opeenvolging van schermen wordt wel de dialoogstruktuur of het transaktieverloop genoemd. Fig. 42.18 geeft een voorbeeld van zo'n dialoogstruktuur. Tijdens de dialoogsimulatie worden de schermen ontworpen die nodig zijn om de gebruiker de transaktie "live" te laten ervaren. Dat hoeft niet te betekenen dat alle mogelijkheden om naar schermen te springen of terug te springen met behulp van funktietoetsen ook gesimuleerd zijn. Verder hoeven bijvoorbeeld niet alle helpschermen gesimuleerd te zijn. Van de gesimuleerde schermen is de opeenvolging vastgelegd in de transaktiedefinitie op de simulator. Die definitie kan worden afgedrukt en vormt de basis voor het ontwerp van de dialoogstruktuur, tesamen met het transaktieschema en de schermlay-outs.
Vervolgens komen we bij de standaards. Het is van belang bij schermontwerp en transaktie-ontwerp bepaalde standaards te handhaven voor schermindeling, het werken met menuschermen, het gebruik van funktietoetsen en dergelijke. Het werk van informatieanalisten, resulterend in schermlay-outs en dialoogontwerp dient gekontroleerd te worden aan de hand van de aanwezige standaards. Wanneer er geen standaards beschikbaar zijn en verscheidene informatie-analisten werken aan het projekt dient de projektleider te zorgen voor de uniformiteit die in het belang is van gebruikers en automatiseerders.
Met het bovenstaande is de gebruikersinbreng in het ontwerpproces in kaart gebracht. Het ontwerp is in overeenstemming met de transaktieschema's in het automatiseringsdossier van de gebruikersafdeling.
Tenslotte komen we bij het onderwerp Transaktie analyse. We zullen nu alleen de pijlen in de figuur behandelen. In het hoofdstuk



Omgeving: Centraal systeem of stand alone decentraal systeem.
Afgebakend projekt, geen netwerkontwerp.
Technische Transaktie analyse.

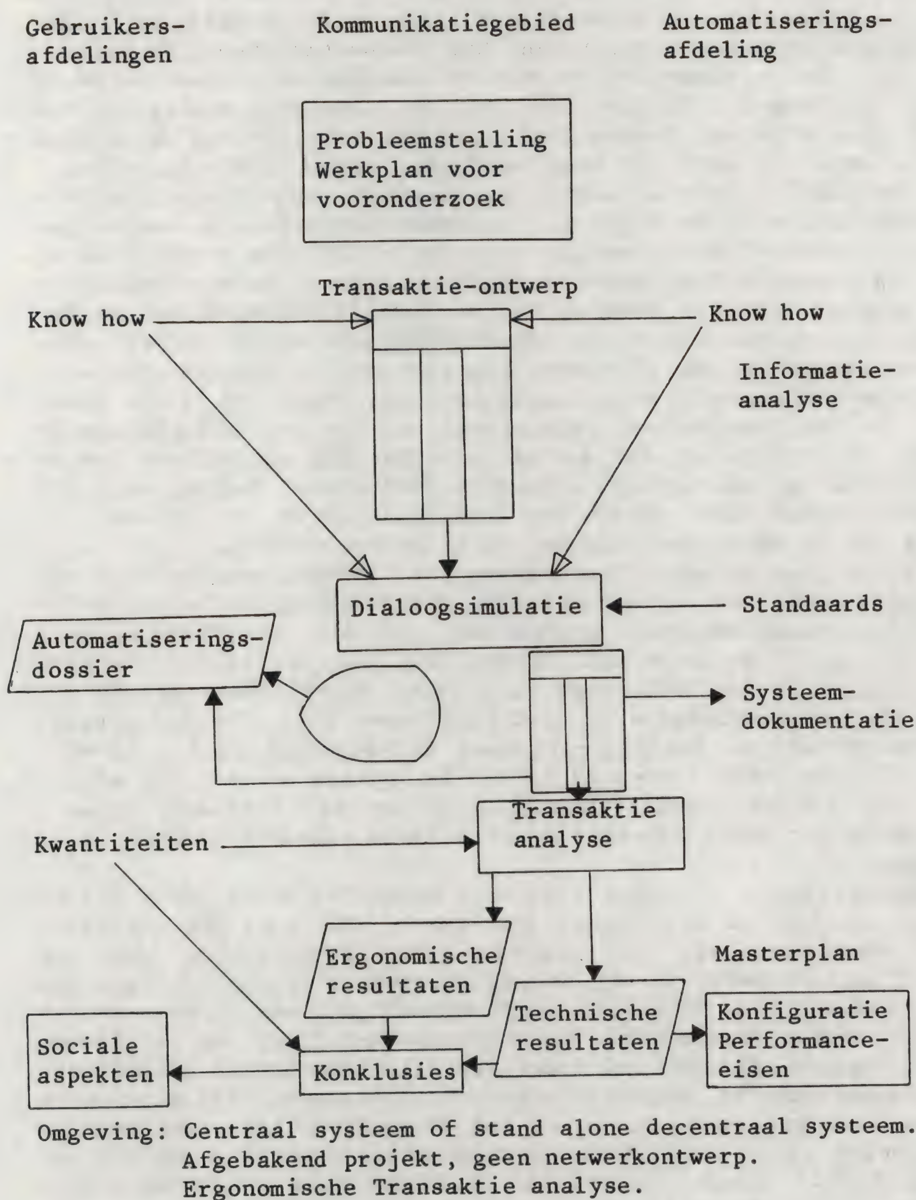
Fig. 42.19 Werkwijze tijdens het technisch ontwerp als vervolg op transaktie-ontwerp tijdens het logisch ontwerp

Transaktie analyse wordt beschreven hoe de resultaten worden verwerkt tot konklusies. Transaktie analyse wordt in principe uitgevoerd door transaktie-analisten. Het transaktieschema, de schermlay-outs en de kwantiteiten binnen transakties vormen de basis voor het transaktiedetailschema van de transaktie-analisten. Met kwantiteiten binnen transakties gaat het bijvoorbeeld om lengtes van velden, de kans dat iets voorkomt, aantal orderregels, etc. Uit Transaktie analyse komen twee pagina's output: de terminaltransaktietijd en de lijn- en responsetijdaspekten. De eerste pagina's van alle geanalyseerde transakties leveren overzichten op met ergonomische resultaten in cijfers. Daaruit kunnen konklusies getrokken worden ten aanzien van de sociale aspekten van de interaktieve toepassingen. De terminaltransaktietijd levert samen met de tweede pagina, de technische resultaten, gegevens ten aanzien van konfiguratie en systeembelasting. Tenslotte leiden kwantiteiten van transakties gekombineerd met de ergonomische resultaten tot konklusies als aantal uren per dag achter een beeldscherm, aantal medewerkers achter beeldschermen. Kortom, de konklusies leiden onder andere tot sociale aspekten in cijfers.

- Fig. 42.19 Werkwijze tijdens het technisch ontwerp.

In deze figuur is het uitgangspunt: het transaktie-ontwerp is begonnen tijdens het logisch ontwerp. Dat betekent dat er een definitief transaktieschema beschikbaar is. Als de scheidingsmuur tussen logisch en technisch ontwerp erg hoog is, zal de transaktie-analist nu het volledige technische detailschema moeten maken. Daarin worden verwerkt: gedetailleerde bestandsbenaderingen, schermlay-outs en hardware-gegevens van beeldschermen en computer. Zijn de muren tussen technisch en logisch ontwerp wat minder hoog dan kan de transaktie-analist gewoon het bestaande detailschema verder uitwerken door er technische gegevens aan toe te voegen.

De twee pagina's met resultaten zijn hetzelfde als tijdens het logisch ontwerp. De technische resultaten leiden nu tot realistische benadering van de responsetijden. Deze worden voor het laatst gelegd naast de nog steeds geldende responsetijdeisen van dialoogsimulatie. De kritische responsetijden kunnen nu voor het laatst worden aangehouden tegen database-ontwerp en programmaspecifikaties. Wanneer met inzet van alle beschikbare optimaliseringsmogelijkheden, bepaalde responsetijden onmogelijk zijn, moet er nu iets gebeuren: nu is er nog gelegenheid om gebruikers te informeren en te laten kiezen uit dezelfde alternatieven als genoemd bij de vorige figuur. Kleine wijzigingen betekenen kleine aanpassingen van het transaktieschema en het detailschema en zorgen voor andere resultaten. Grote wijzigingen moeten waarschijnlijk via dialoogsimulatie doorgenomen worden met de gebruiker. Nieuwe transakties worden op precies dezelfde manier ontworpen



Omgeving: Centraal systeem of stand alone decentraal systeem.
Afgebakend projekt, geen netwerkontwerp.
Ergonomische Transaktie analyse.

Fig. 42.20 Transaktie-ontwerp tijdens het vooronderzoek

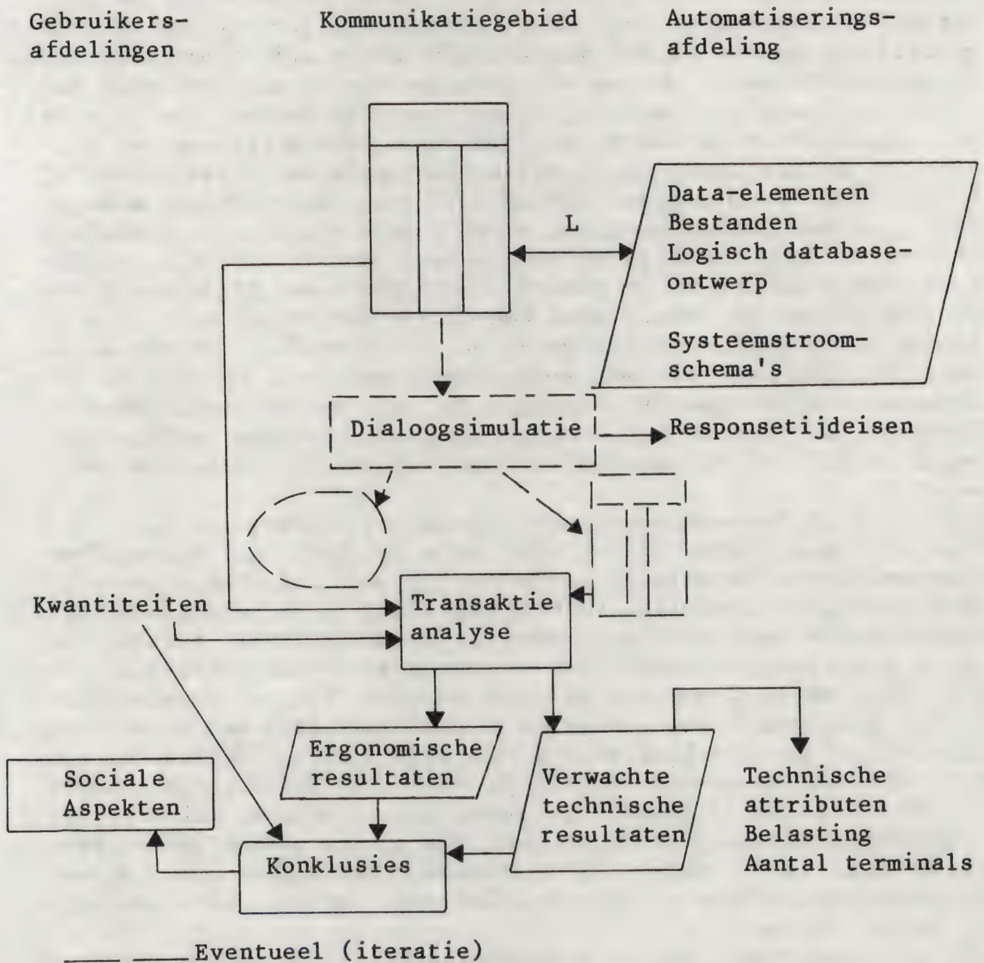
als tijdens het logisch ontwerp. In alle drie gevallen worden de transaktieschema's die ontstaan zijn, door de gebruiker in zijn automatiseringsdossier opgenomen. Behalve ten gevolge van de responsetijden kunnen er nog aanpassingen nodig zijn tengevolge van de gekozen hardware. Zo kan het gebeuren dat er beeldschermen beschikbaar komen die meer of minder funkties hebben dan tijdens het logisch ontwerp bekend was. Ook deze verschillen kunnen leiden tot aanpassingen van transaktieschema's en detailschema's. Zoals reeds is aangegeven in het deel voor het taktisch management, is het onverstandig de muren tussen logisch en technisch ontwerp zo hoog mogelijk op te trekken, dat dit soort iteraties niet meer mogelijk is. De meeste ontwerpprocessen zijn iteratieve processen! Aan de andere kant behoort de ontwerper niet in zijn manier van werken te anticiperen op die iteraties. In situaties waar de apparatuur al lang gestandaardiseerd is, behoort de informatie-analist goed op de hoogte te zijn van de mogelijkheden. Dan moet het niet zo zijn dat hij de gebruiker maar wat toezegt, omdat er toch wel terugkoppeling komt tijdens het technisch ontwerp.

- Fig. 42.20 Transaktie-ontwerp tijdens het vooronderzoek.

Zoals is aangegeven in het deel voor het taktisch automatiseringsmanagement bestaat er een wezenlijk verschil tussen prototyping en dialoogsimulatie. Voor de gebruiker is er weinig verschil tussen beide methoden, maar voor de automatiseerder verschillen ze zo sterk, dat ze nauwelijks vergelijkbaar zijn. Omdat dialoogsimulatie wordt uitgevoerd met een portable P.C, er geen bestanden of programma's nodig zijn en de gebruiker toch een uitstekend idee krijgt wat een beeldscherm voor zijn werk kan betekenen, kan het transaktie-ontwerp eigenlijk op elk willekeurig moment plaatsvinden. En in sommige gevallen kan dat moment wel eens het vooronderzoek zijn. Uit het oogpunt van projektaanpak is het zeer ongebruikelijk. In de meeste systeemontwikkelingsmethoden is het vooronderzoek echter zo vaag gedefinieerd, dat transaktie-ontwerp er altijd bij kan.

Bij de toelichting van de vorige figuren is aangegeven hoe het transaktie-ontwerp gekoppeld wordt aan de overige ontwerpaktiviteiten. Wanneer die synchronisatie gehandhaafd blijft, mag het transaktie-ontwerp best plaatsvinden tijdens het vooronderzoek. Daar moet dan wel een duidelijke reden voor bestaan. Enkele praktijksituaties:

- De direktie van een sterk gedecentraliseerd bedrijf met kleine vestigingen overweegt te gaan automatiseren. Van te voren staat vast dat automatisering alleen zin heeft als de vestigingen informatie kunnen uitwisselen. Het daarvoor te bouwen netwerk zou wel eens een te groot deel van het budget kunnen opslokken, dus wil men tijdens vooronderzoek een netwerkontwerp gebaseerd op



Omgeving: Centraal systeem of stand alone decentraal systeem.
Afgebakend projekt, geen netwerk ontwerp.
Logische Transaktie analyse.

Fig. 42.21 Werkwijze tijdens logisch ontwerp, als vervolg op transaktie-ontwerp tijdens het vooronderzoek

Transaktie analyse en dus op transaktie-ontwerp.

- Een bedrijf heeft tijdens het vooronderzoek volgens een systeemontwikkelingsmethode de te automatiseren procedures in kaart gebracht. Van iedere funktionaris is in principe vastgesteld wat hij straks gaat doen via een beeldscherm. Door de veelheid van transakties is iedereen een beetje het overzicht kwijt, en enkelen beginnen zich af te vragen, hoeveel tijd er voor sommige funktionarissen nog overblijft voor "het normale werk". Transaktie-ontwerp levert de cijfers.
- Een bedrijf wil lokettransakties gaan uitvoeren met behulp van beeldschermen. Er moet, voordat geld besteed wordt aan detailanalyses en -ontwerp, vaststaan dat de beeldschermtransakties zo vlot verlopen dat de wachtrijen bepaalde lengtes niet overschrijden. Transaktie-ontwerp levert de cijfers die nodig zijn voor de berekeningen volgens de wachtrijtheorie.

Gebruikers en informatie-analist beginnen aan het transaktie-ontwerp. Het enige waar de informatie-analist eventueel al rekening mee kan houden zijn standaards, bijvoorbeeld ten aanzien van de schermindeling. Verder verloopt het transaktie ontwerp op de normale manier. De kwantiteiten die nodig zijn voor de ergonomische Transaktie analyse kan de gebruiker net zo goed nu aangeven als tijdens het logisch ontwerp.

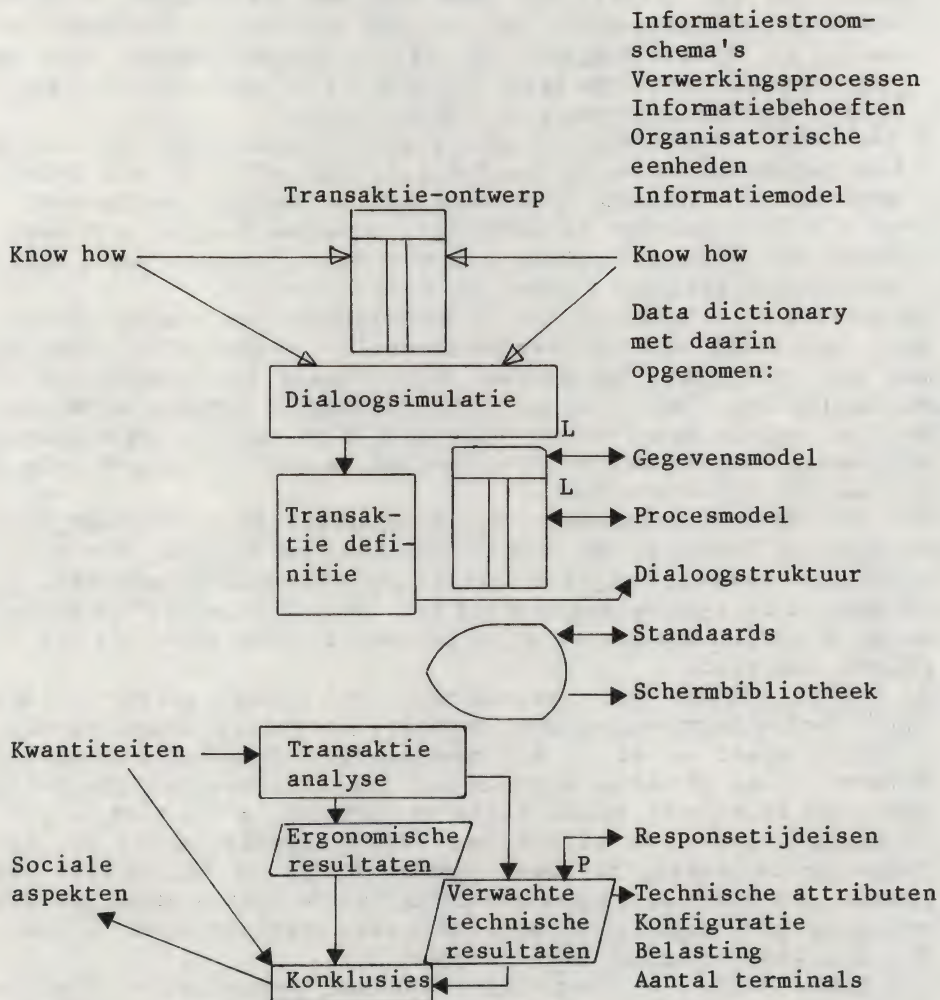
De resultaten en konklusies zijn uiteraard direkt bruikbaar voor de sociale aspekten. Het masterplan wordt nu uitgebreid met details die normaal pas tijdens het logisch ontwerp ontstaan. Ze blijven in de systeemdossiers tot het logisch ontwerp. Dan worden ze op de eerder beschreven wijze behandeld. Tenminste, als het projekt doorgaat.

Nu hebben tijdens het vooronderzoek de eindgebruikers al met "hun" beeldschermtoepassingen gewerkt, de sociale aspekten zijn gekwantificeerd en er is een nauwkeurig kostenplaatje gemaakt. Wanneer nu de direktie besluit het projekt niet door te laten gaan, dan is er niet geinvesteerd in hardware, zelfs niet in prototyping hardware, er zijn alleen kosten gemaakt om tot een gefundeerde beslissing te komen. Verder blijkt in de praktijk dat wanneer een planning en een schatting van de kosten wordt gemaakt op grond van Transaktie-ontwerp, die veel betrouwbaarder is dan de gebruikelijke natte-vingermethoden.

Samengevat kunnen we vaststellen dat het vooronderzoek niet de aangewezen fase is om transakties te ontwerpen, hoewel er echter heel duidelijke redenen kunnen zijn om het toch te doen. Voor de gebruiker kan het net zo goed tijdens het vooronderzoek, als tijdens het logisch ontwerp. Het levert een uitstekende basis op voor de planning van het logisch ontwerp.

- Fig.42.21 Werkwijze tijdens het logisch ontwerp.

Wanneer tijdens het vooronderzoek de transakties zijn ontworpen



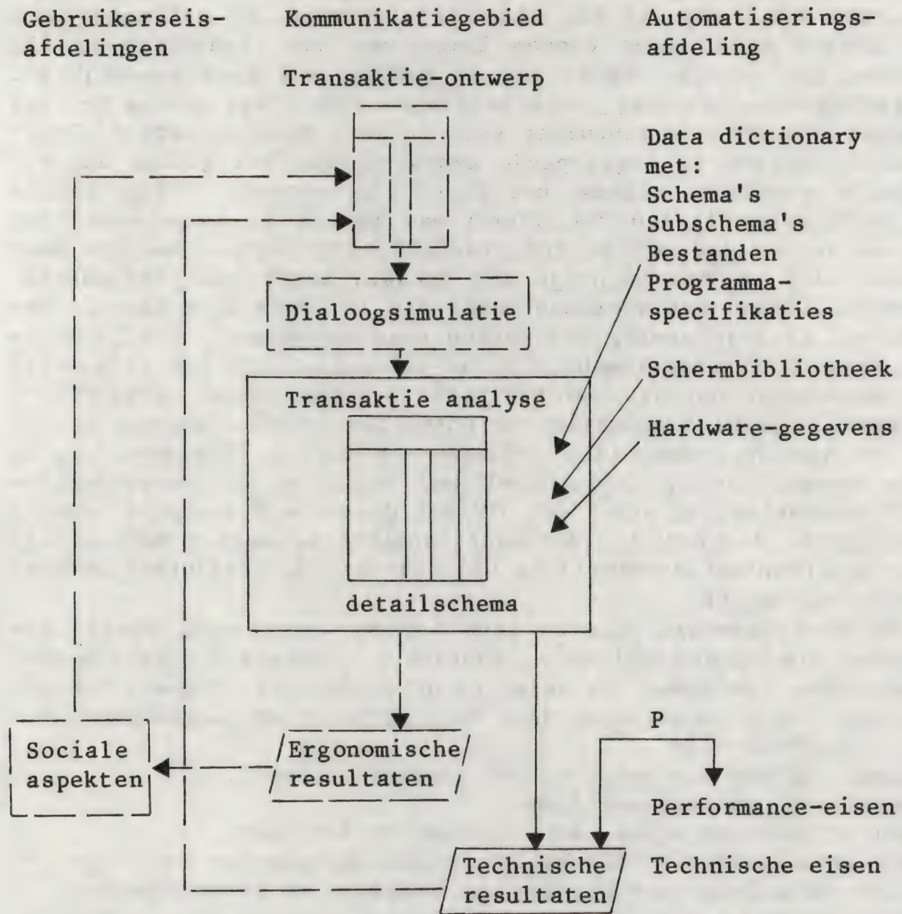
Omgeving: Centraal systeem of stand alone decentraal systeem.
Centrale database en data dictionary.
Centrale ontwikkeling en beheer.
Geen netwerkontwerp. Logische Transaktie analyse.

Fig. 42.22 Transaktie-ontwerp tijdens het logisch ontwerp.

behoeven in het logisch ontwerp alleen nog de kontroles te worden uitgevoerd die ook bij Fig 42.17 zijn genoemd. Op dat moment zou het nadeel naar voren kunnen komen van het transaktie-ontwerp tijdens het vooronderzoek. Het is immers heel goed mogelijk dat de informatie-analisten, niet gehinderd door enige kennis van het logisch ontwerp, de gebruiker veel te veel beloofd hebben. Daardoor is de kans op ingrijpende aanpassingen veel groter dan bij transaktie-ontwerp tijdens het logisch ontwerp en vandaar dat in de gestippelde lijnen het iteratieve proces is aangegeven. Dat kan betekenen dat een aantal transakties opnieuw ontworpen moet worden: dat is dan de prijs die betaald wordt voor transaktie-ontwerp tijdens het vooronderzoek. Als iedereen zich daarvan bewust is, is deze manier van werken best akseptabel. Het grootste deel van de transaktieschema's, de schermlay-outs, de resultaten van Transaktie analyse zal bruikbaar blijven, omdat uiteindelijk de gebruikers de transakties zo ontworpen hebben. Wanneer men 30 of 40% van de transakties opnieuw zou moeten ontwerpen, is er iets vreemds aan de hand. Ofwel met de mening van de gebruikers over automatisering ofwel met die van de automatiseerders over de gebruikers. Dat had de informatie-analist op moeten merken tijdens het transaktie-ontwerp en hij had daar de projektleider over moeten informeren!
In de bovengenoemde figuren ging het om omgevingen, waarin begrippen als data dictionary, modelbouw, systeemontwikkelingsmethoden niet voorkomen. We zulen nu dezelfde serie figuren behandelen maar dan in omgevingen waar deze methoden en gereedschappen wel in gebruik zijn.
De omgeving kan als volgt worden gekarakteriseerd:

- een groot centraal mainframe
- een centrale database, data dictionary/directory
- een grote centrale ontwikkelingsafdeling, waarvan een groep zich bezig houdt met standaards, methoden en gereedschappen. In de termen van de paragraaf Methoden en omgeving, een M3-Cx-omgeving.

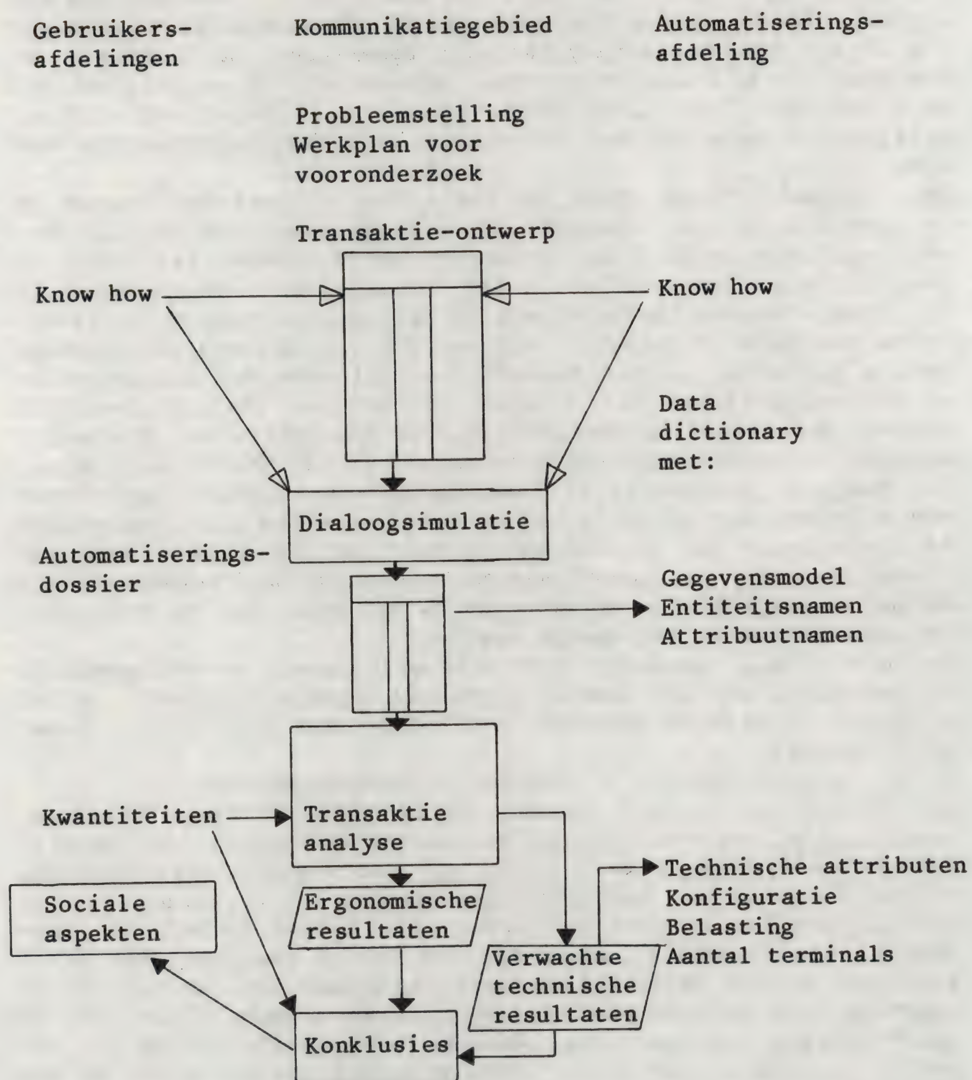
In deze omgeving kunnen nieuwe toepassingen worden ontwikkeld voor het mainframe of voor stand alone decentrale systemen.
- Fig. 42.22 Transaktie-ontwerp tijdens het logisch ontwerp.
Tijdens het vooronderzoek zijn organisatorische eenheden, informatiebehoeften en verwerkingsprocessen in kaart gebracht, informatiestroomschema's en informatiemodellen gemaakt. Dan is er al veel gekommuniceerd met gebruikers, maar alleen ten dienste van de analyse van het bedrijfsgebeuren. Op basis van het informatiemodel kunnen nu samen met de gebruikers transakties worden ontworpen. De definitieve transaktieschema's worden vergeleken met de inmiddels ontwikkelde gegevens- en procesmodellen. Het transaktieschema blijft een gebruikersdokument en zal nooit geschreven



Omgeving: Centraal systeem of stand alone decentraal systeem.
Centrale database en data dictionary.
Centrale ontwikkeling en beheer.
Geen netwerkontwerp.
Technische Transaktie analyse.

Fig. 42.23 Werkwijze tijdens het technisch ontwerp als vervolg op transaktie-ontwerp tijdens het logisch ontwerp.

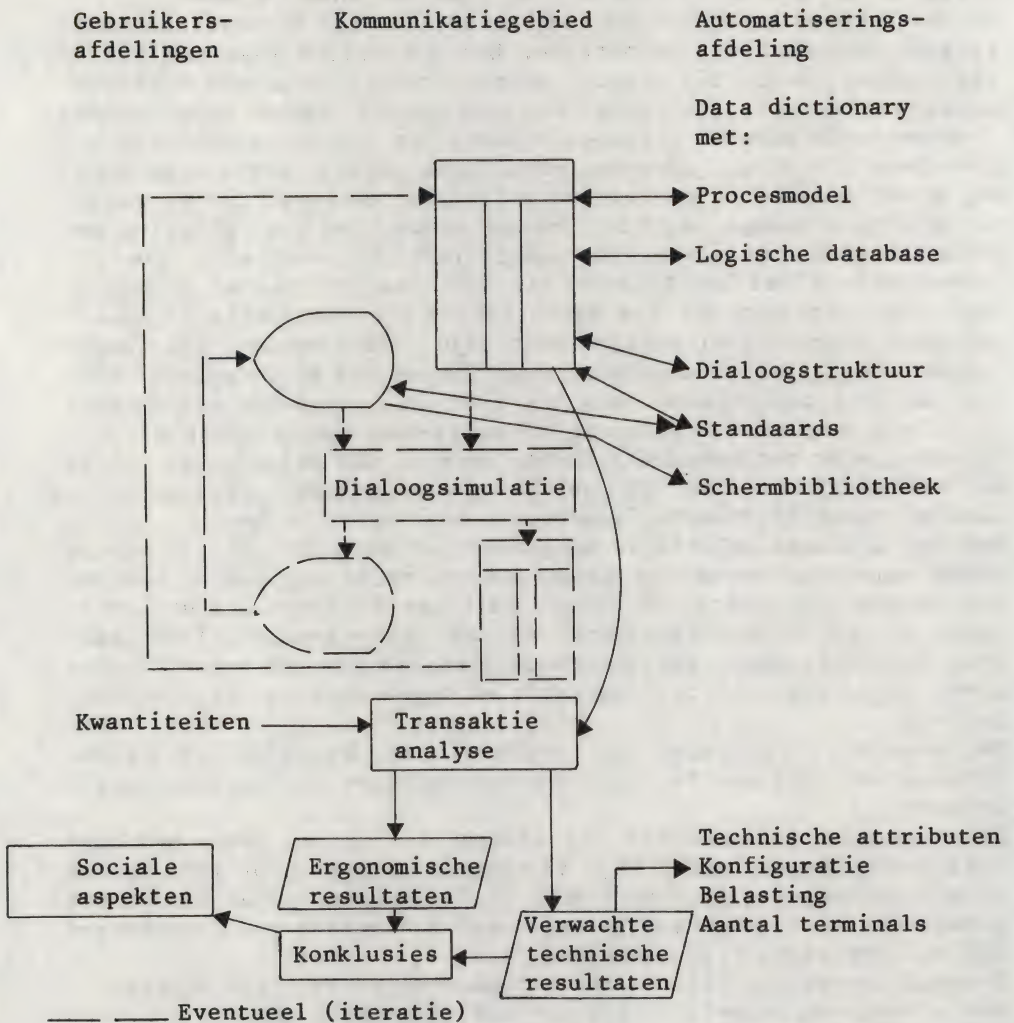
zijn in de taal van de modellen. De tekst op het detailschema kan echter volledig worden aangepast aan de standaardnaamgeving zoals die in een data dictionary kan zijn opgeslagen. Dat betekent dat het mogelijk is het detailschema "automatisch" te vergelijken met de dictionary. Bij grote aantallen transakties en nog veel te ontwikkelen projekten kan de bouw van zo'n vergelijkingsprogramma zinvol zijn.
Het transaktieschema vormt de basis voor de dialoogstruktuur of de opeenvolging van schermen. Ook deze dialoogstruktuur kan worden opgeslagen in de data dictionary. De transaktie kan worden opgenomen in de data dictionary als entiteitstype, de ergonomische en technische resultaten vormen de attributen. Wanneer de resultaten van alle Transaktie analyses in een bestand of database worden opgenomen, kunnen bezettingsoverzichten van beeldschermen of belastingcijfers van groepen transakties snel geproduceerd worden. Het handmatig verwerken van de resultaten van Transaktie analyse tot overzichten wordt behandeld in het hoofdstuk Transaktie analyse. Daarmee is dan een exakte specifikatie gegeven van een programma dat op basis van de resultaten van alle transakties de overzichten kan produceren. Wanneer er nog verscheidene onlineprojekten gerealiseerd moeten worden, is het verstandig een dergelijk programma te ontwikkelen op het mainframe of op de microcomputer van de informatie-analist.
De in de figuur genoemde technische attributen, de konfiguratie, de belasting en het aantal terminals worden behandeld in het hoofdstuk Transaktie analyse. In dit hoofdstuk gaat het om transaktie-ontwerp.
- Fig. 42.23 Werkwijze tijdens het technisch ontwerp.
Als tijdens het logisch ontwerp het transaktie-ontwerp is uitgevoerd, is er een detailschema van iedere transaktie. Dat detailschema bevatte gegevens die tijdens het logisch ontwerp beken waren. Tijdens het technisch ontwerp is de struktuur van de fysieke database bekend en verder alle andere technische gegevens: soort terminals, soort netwerk, soort computer. Dat betekent dat nu het detailschema wordt aangepast aan de technische omgeving van de transakties. Het rekenprogramma levert nu de nauwkeurigste resultaten op. Wanneer bijvoorbeeld blijkt dat bepaalde responsetijden nooit gehaald zullen worden is er nu nog gelegenheid kontakt op te nemen met de informatie-analist en de gebruikers. Zo kunnen er vele oorzaken zijn voor een terugkoppeling met de gebruiker, resulterend in aanpassingen van het oorspronkelijke ontwerp. Ontwerpen is een iteratief proces; hier worden de wegen aangegeven voor de iteraties.
Samenvattend kunnen we vaststellen dat, afgezien van iteratieve akties, tijdens het technisch ontwerp het aksent ligt op Transaktie analyse.



Omgeving: Centraal systeem of stand alone decentraal systeem.
Centrale database en data dictionary.
Centrale ontwikkeling en beheer.
Geen netwerkontwerp.
Ergonomische Transaktie analyse.

Fig. 42.24 Transaktie-ontwerp tijdens het vooronderzoek.

- Fig. 42.24 Transaktie-ontwerp tijdens het vooronderzoek.
In dit verband wordt met vooronderzoek een fase bedoeld die vooraf gaat aan het logisch ontwerp. Een dergelijke fase kan heten: analysefase, definitie studie, vooronderzoek, corporate analysis, detailed analysis. Er moet een duidelijke reden zijn om aan transaktie-ontwerp te beginnen tijdens het vooronderzoek. Ook bij bedrijven die reeds jaren met mainframes werken, wordt soms overwogen een projekt te realiseren met microcomputers. Een dergelijke oplossing vraagt een heel andere aanpak dan een oplossing met beeldschermen en een clustercontroller. Aan het eind van het vooronderzoek zal er toch zoiets als een kostenplaatje moeten zijn. Dat betekent dat een keuze tussen microcomputers en beeldschermen eigenlijk al gemaakt moet zijn. Zeker wanneer de eindgebruikers om allerlei andere redenen liever een microcomputer hebben dan een beeldscherm. Als het gaat om gekoppelde microcomputers, een centrale fileserver of multi-user-micro's zal de performance weer het bekende probleem worden. Dat betekent dat er in de fase waarin het koncept voor de oplossing wordt gekozen, behoefte is aan cijfermateriaal over belasting en verkeer.
Wanneer dialoogsimulatie is uitgevoerd, kunnen afhankelijk van de reeds aanwezige corporate metagegevens, reeds een aantal kontroles worden uitgevoerd. Te denken valt aan de vergelijking van de tekst op het transaktieschema met het gegevensmodel, funktiemodel, entiteitsnamen, attribuutnamen, standaards. De schermlay-out wordt vergeleken met de standaards en opgeborgen tot het logisch ontwerp.
De gebruikers ontvangen een kopie van de schermlay-out en het transaktieschema met de opmerking dat het gaat om een voorlopig ontwerp.
De ergonomische resultaten van Transaktie analyse geven aan hoeveel beeldschermen nodig zijn of hoeveel tijd de gebruiker achter zijn microcomputer zit. De technische resultaten geven aan of het geheel met microcomputers gerealiseerd kan worden qua hoeveelheid I/O's, verkeer en responsetijden.
Transaktie-ontwerp tijdens het vooronderzoek biedt een stevige basis voor een vroegtijdige keuze uit alternatieven.
- Fig. 42.25 Werkwijze tijdens het logisch ontwerp als vervolg op het transaktie-ontwerp tijdens het vooronderzoek.
Nu aan het begin van het logisch ontwerp alle transaktieschema's of een groot deel ervan beschikbaar zijn, kan daarvan bij het ontwerpen van de gegevens- en funktiemodellen gebruik worden gemaakt. De schermlay-outs van de dialoogsimulator worden nu beschikbaar gesteld aan de ontwerpers.
De iteratie, die met stippellijnen is aangegeven, zal nu meestal moeten worden uitgevoerd. Tijdens het vooronderzoek ging het im-



Omgeving: Centraal systeem of stand alone decentraal systeem.
Centrale database en data dictionary.
Centrale ontwikkeling en beheer.
Geen netwerkontwerp.
Logische Transaktie analyse.

Fig. 42.25 Werkwijze tijdens het logisch ontwerp als vervolg op transaktie-ontwerp tijdens het vooronderzoek.

mers om de grote lijnen en ruwe schattingen. Bovendien zit er tussen vooronderzoek en logisch ontwerp meestal een tijdverschil van enkele tot vele maanden. Als het goed is heeft dialoogsimulatie de gebruikers aan het denken gezet. Aan de kant van de automatisering zijn inmiddels ook wat andere plannen ontstaan. Nu er een konceptkeuze is gedaan, zijn de grenzen voor nieuwe plannen uiteraard beperkt. Alles moet nu met de gekozen middelen gerealiseerd worden. Transaktie-ontwerp of iteratieve akties houden de vinger aan de pols.
Transaktie analyse wordt nu gedetailleerder voorzover het gaat om de verwerkingsaspekten. De transaktie-analist moet de detailschema's aanpassen aan de situatie zoals die nu vastligt in het gegevens- en funktiemodel. De resultaten van deze analyses worden gelegd naast de resultaten van het vooronderzoek. Duidelijke afwijkingen moeten binnen het gekozen koncept verwerkt worden in het ontwerpproces. Het kan natuurlijk gebeuren dat de inzichten inmiddels zo veranderd zijn dat het gekozen koncept niet meer bruikbaar is, maar tegen dergelijke situaties is natuurlijk geen enkele ontwerpmethode opgewassen. In ieder geval zijn de argumenten nu gekwantificeerd en dus goed bespreekbaar. De stuurgroep mag beoordelen of de nieuwe plannen worden gerealiseerd of dat men de konceptkeuze handhaaft. In principe zou het tijdens het logisch ontwerp nog mogelijk moeten zijn over te schakelen op een andere hardware-oplossing.

42.5 Geografie in analyse en ontwerp

De geografie van een bedrijf is in verband met de automatisering van belang voor een aantal aspekten:
- het netwerk
- de lokatie van computers
- de lokatie van gegevens.

In een C1-omgeving gaat het om een lokatie met een computer en de gegevens. In een C2-omgeving is de lokatie van de gegevens gelijk aan die van de computers: de gegevens zijn opgeslagen waar ze gebruikt worden. In een C2N-omgeving gaat het om een netwerk tussen de computers en zullen gegevens op verscheidene plaatsen gebruikt worden, terwijl ze op een of meer plaatsen zijn opgeslagen. Als een C2N-omgeving ontworpen moet worden, zal het netwerkkoncept afhangen van het verschil tussen opslag en gebruik van gegevens. Het gebruik van gegevens is in kaart gebracht wanneer alle transakties zijn ontworpen en geografisch bepaald via de werkplekken. De lokatie van de gegevensopslag is nu de belangrijkste parameter in het netwerkontwerp. Een technische Transaktie analyse levert cijfers op voor het verkeer tussen lokaties. Via een aantal van dergelijke analyses met de lokatie van de gegevensopslag als pa-

Geografische eenheden	Toelichting	Netwerk
Vestigingsplaats	Plaatsnaam	WAN
Vestiging, gebouw, kantoor	Adres	Lokaal netwerk
Werkplekgroep	Afdeling, verdieping	LAN, cluster-controllers
Werkplek	Bureau, tekentafel	Netwerkaansluiting

Fig. 42.26 Geografische eenheden en netwerken.

rameter kunnen enkele netwerkkoncepten worden ontwikkeld en doorgerekend. Deze aanpak maakt deel uit van de netwerkontwerp-methoden die behandeld wordt in het deel voor de transaktie-analist.
Hetzelfde geldt voor de C3N-, C4N- en C5N-omgevingen.
De informatie-analist moet behalve de transaktie ontwerpen ook hun lokatie vaststellen. Enerzijds moeten dus de werkplekken met beeldschermen geografisch in kaart worden gebracht, anderzijds moet per transaktie worden vastgesteld op welke werkplekken die wordt uitgevoerd. Zo ontstaat het algemene attribuut: lokatie(s).
Het geografische plaatje van een bedrijf kan zeer komplex zijn, maar ook erg simpel. In Fig. 42.26 wordt een bedrijf gesplitst in geografische eenheden. Elk niveau onderscheidt zich van een ander niveau door een ander soort netwerk.
Vestigingsplaatsen worden onderling verbonden door datakommunikatielijnen. Meestal zijn dat interlokale lijnen. Men spreekt in de Engelstalige literatuur over een wide area network (WAN). Per vestigingsplaats kunnen verschillende vestigingen voorkomen: kantoren, fabrieksgebouwen. Karakteristiek is dan het adres: de straat en het nummer. Dergelijke vestigingen kunnen via lokale PTT-lijnen met elkaar verbonden worden.
Per gebouw kunnen groepen werkplekken worden vastgesteld, meestal bepaald door organisatorische aspekten: afdelingen, sekties, groepen e.d. Binnen zo'n gebouw kunnen de groepen van een local

area network (LAN) met elkaar of met hun computer verbonden zijn. De verbindingen kunnen ook gerealiseerd zijn met behulp van clustercontrollers.
Tenslotte komen we op het niveau van de werkplek. Daar bevindt zich het beeldscherm en de aansluiting aan het netwerk.
De lokatie van een werkplek wordt dus maximaal aangegeven met: plaats.vestiging.werkplekgroep.werkplekaanduiding.
Bij bedrijven met veel vestigingsplaatsen, maar slechts een vestiging per plaats, kan het tweede element in de lokatie natuurlijk vervallen.
Wanneer zo alle werkplekken in kaart zijn gebracht en van iedere transaktie de lokatie(s) zijn aangegeven kunnen, al dan niet geautomatiseerd, overzichten worden gemaakt van transakties per gebouw of per vestigingsplaats. Per transaktie is ook bekend welke gegevens gebruikt worden. Deze koppeling wordt tot stand gebracht via de transaktieschema's of door bij ieder gegeven op te nemen in welke transakties het wordt gebruikt. Wanneer men beschikt over een datadictionary/directory kan men opvragen op welke lokaties een gegeven of een gegevensverzameling gebruikt wordt. Zo ontstaat een beeld van het gegevensgebruik. Dit gebruik gekoppeld aan een lokatie voor de opslag, levert een netwerkkoncept. Deze aanpak maakt deel uit van de netwerkontwerpmethode die behandeld wordt in het deel voor de transaktie-analist. Doel van deze paragraaf is alleen aangeven dat de geografische situatie in kaart gebracht moet worden en waarom.
Rest nog de vraag wanneer het moet gebeuren. Wanneer de informatie-analist de aktiviteiten en procedures in kaart brengt zou hij per procedure de lokatie kunnen aangeven, op de bovenomschreven manier. Later worden procedures omgezet in transakties en dan zijn de lokaties per transaktie ook bekend. Hij zou ook kunnen wachten tot de transakties ontworpen zijn en dan per transaktie de lokatie(s) aangeven.
Hij zou ook zowel bij de analyse als bij het ontwerp de lokatie(s) aan kunnen geven. Voor het netwerkontwerp is alleen van belang dat per transaktie de lokaties bekend zijn.
Voor het ontwerp van het netwerk is het uiteraard niet van belang wanneer de geografie van de werkplekken in kaart is gebracht. In het algemeen zal tijdens de informatie-analyse al iets bekend worden over de geografie in grote lijnen. Als informatie-analyse wordt uitgevoerd met een geautomatiseerd gereedschap, (20) kan alle verkregen informatie worden vastgelegd op een manier die bruikbare gegevens oplevert voor volgende fasen.
Opgeslagen moeten worden:
- alle werkplekken met hun geografische aanduiding,
- alle procedures per werkplek,
- alle funktionarissen en hun werkplekken.

Tijdens het logisch ontwerp ontstaan transakties. Een transaktie kan nul, een of meer procedures vervangen. Nu worden opgeslagen:
- alle transakties per werkplek,
- de vervallen procedures per transaktie,
- alle gegevensverzamelingen per transaktie,
- alle attributen van het entiteitstype transaktie.

Een gereedschap dat krachtig genoeg is kan nu een aantal overzichten maken. Voor het netwerkontwerp gaat het om een overzicht dat aangeeft welke gegevens op iedere vestigingsplaats worden gebruikt, inklusief frekwenties en pieken. Daarmee is de geografie van het gegevensgebruik in kaart gebracht en kan een begin gemaakt worden met het ontwerp van een gedistribueerde database. In het deel voor de transaktie-analist wordt verder uitgewerkt hoe het netwerkontwerp verloopt.
Naast het genoemde overzicht zijn er natuurlijk nog wat overzichten te maken waarin personeelszaken en ondernemingsraad interesse zullen hebben. Er kan een overzicht worden gemaakt dat aangeeft welke procedures per afdeling of per vestiging zijn vervallen of een overzicht dat aangeeft hoeveel uur per dag per afdeling er met beeldschermen wordt gewerkt.
Het is meestal een probleem om van de gebruikersorganisatie voldoende tijd te krijgen voor informatie-analyse en systeemanalyse, maar het uitzicht op dit soort overzichten kan best een bijdrage leveren in de diskussie.

42.6 Transaktie-ontwerp en beeldschermontwerp

Het ontwerpen van interaktieve toepassingen lijkt te moeten beginnen met het maken van beeldschermontwerpen. De eerste dokumenten die aanwezig zijn in de projektdokumentatie zijn dan ook meestal de schermlay-outs. Vanuit de historie bezien is dat ook begrijpelijk. Automatiseerders hielden zich immers bezig met bestanden en programma's. Zo'n beeldscherm is eigenlijk ook niets anders dan een bestand waar afwisselend van wordt gelezen en naar wordt geschreven. Is de schermlay-out bekend dan zijn de records en de items bekend en kan er aan programma's gedacht worden. Vroeger was de enige interface met de gebruiker de printoutput. Nu verschijnt een deel van de printoutput op het scherm en de operator mag nog iets intypen ook!
Zoals vroeger de lay-out van de printoutput werd besproken met de gebruiker, zo wordt nu de schermlay-out met hem doorgenomen. Er zijn zelfs ontwerpers die beweren dat er eigenlijk geen verschil is tussen batch-verwerking en interaktieve verwerking. Wat vroeger op een ponskoncept werd gezet, wordt nu ingetypt en wat vroeger op de printer verscheen wordt nu op het beeldscherm gezet. Zo eenvoudig is dat.

Interaktieve transakties kunnen vanuit verschillende standpunten worden bezien. Vanuit het automatiseringsstandpunt zoals bovenomschreven, bezien, is het beeldscherm een van de vele I/O apparaten dat door een programma bestuurd moet worden. Vanuit het standpunt van de gebruiker bezien, is het een nieuw gereedschap dat hij moet inpassen in zijn overige werkzaamheden.
Personeelszaken en de ondernemingsraad zien er een stuk taakverandering in met nogal onvoorspelbare gevolgen.
Hoe men het ook beziet, iedereen zal het er over eens zijn dat transakties met beeldschermen moeten worden ingepast in het geheel van aktiviteiten van de gebruiker. Wanneer er orderentry gedaan wordt via beeldschermen dan komen de orderbonnen ergens vandaan en gaan ergens naar toe. De kontroles worden anders dan in de handmatige situatie, omdat het computersysteem een aantal kontroles overneemt. Sommige handmatige aktiviteiten zullen helemaal wegvallen. Kortom, het ontwerpen van transakties behoort te beginnen bij deze aspekten. Het transaktie-ontwerp beschrijft het geheel van handelingen door de gebruiker uit te voeren. Het geheel wordt vastgelegd op een transaktieschema, een document dat voor gebruikers, personeelszaken en de ondernemingsraad prima leesbaar is. Op welke plaats op het beeldscherm de gegevens verschijnen of moeten worden ingetypt is in dit stadium volkomen onbelangrijk.
Per werkplek is de volgorde dan ook:
- Vaststelling van de benodigde transakties.
- Transaktie-ontwerp.
- Dialoogontwerp.
- Beeldschermontwerp.

Zoals meestal, gaat het ook hier om een iteratief proces. Wanneer gebruik gemaakt wordt van dialoogsimulatie dan zijn de iteraties binnen korte tijd realiseerbaar. Dat betekent, uiteraard afhankelijk van de komplexiteit van de transakties en het aantal gebruikers dat erbij betrokken is, dat binnen enkele dagen per werkplek is vastgesteld welke transakties nodig zijn en hoe die er uit moeten zien. Volledigheidshalve dient te worden opgemerkt dat er best enige maanden kunnen verlopen tussen transaktie-ontwerp en beeldscherm-ontwerp. Dat zal zeker het geval zijn wanneer men standaards wil ontwikkelen voor schermlay-out en bediening van het toetsenbord. Onafhankelijk van de tijdsaspekten is het echter wezenlijk transaktie-ontwerp als een separate aktiviteit in te plannen die voorafgaat aan het beeldschermontwerp.
Transaktie-ontwerp is trouwens niet alleen in het belang van gebruikers. De menselijke handelingen zijn van grote invloed op de belasting van het systeem. Daarom alleen al zouden automatiseerders eerst transakties moeten ontwerpen en later de er nu eenmaal bijhorende schermlay-outs.

42.7 Transaktie-ontwerp in distributieve omgevingen

In distributieve omgevingen gaat het om het gebruik van elders opgeslagen gegevens. Naast de behandelde aspekten van transaktie-ontwerp komen er dan nog enkele problemen bij.
Met name in distributieve omgevingen is het van belang dat de informatie-analist bedenkt dat het ontwerpen van transakties inhoudt dat een funktionaris op een bepaalde manier omgaat met gegevens. Het aanmaken, wijzigen, opvragen en verwijderen van gegevens via transakties moet passen bij de funktie. Vandaar dat nieuwe taak/funktie-omschrijving tijdens het logisch ontwerp gemaakt moeten worden. Transakties worden uitgevoerd op werkplekken, die geografisch zijn bepaald, zoals is aangegeven in de vorige paragraaf. Als op de transaktieschema's goed is aangegeven om welke gegevens of gegevensverzameling het gaat, is de topologie van het gebruik van gegevens daarmee vastgelegd.
Het ontwerpen van distributieve systemen is zo ingewikkeld omdat gegevens op diverse plaatsen kunnen worden opgeslagen. Opslag op de plaats van gebruik levert in ieder geval de kortste responsetijden op. Als dezelfde gegevens op verschillende plaatsen worden opgeslagen is er alleen al een netwerk nodig om ze konsistent te houden.

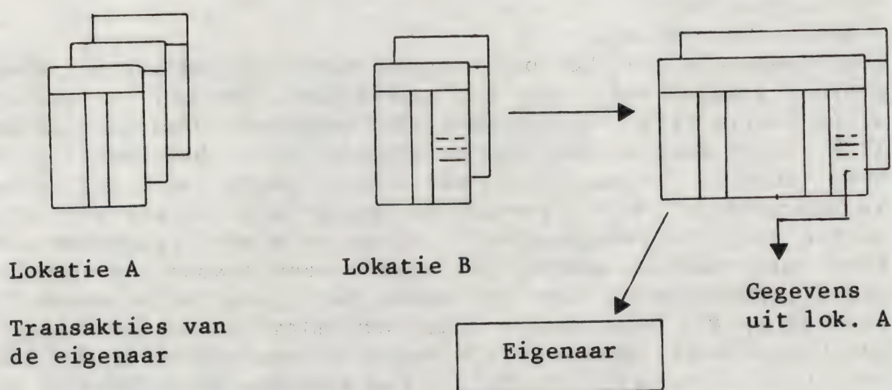


Fig. 42.27 Decentrale transaktieschema's voor de eigenaar van gegevens.

Voor gebruikers is de opslag meestal transparant. Daarom zou het hele ontwerp het terrein van de informatie-analist kunnen vallen. Hij levert transaktieschema's af en daarmee is voor hem de kous af. Uitzondering op die regel is de situatie waarin de gebruiker, bijvoorbeeld de eigenaar van een gegevensverzameling, wil dat beschreven wordt wat er op andere lokaties aan transakties is ontworpen die zijn gegevens gebruiken. In zo'n geval kunnen de betrokken transaktieschema's worden uitgeleid tot decentrale transaktieschema's, zie Fig. 42.27. In principe is het een zaak van implementatie of gegevens bij de gebruiker of elders worden opgeslagen, maar het kan geen kwaad als de informatie-analist dit soort transaktieschema's maakt: ze moeten tijdens het technisch ontwerp van een distributief systeem toch worden gemaakt.
Vanuit het logisch ontwerp wordt met de volgende punten een bijdrage geleverd in het ontwerp van het netwerk in een distributieve omgeving:

- transaktieschema's, centraal of decentraal, die qua gegevensbeheer goed op de gebruikers zijn afgestemd,
- transaktieschema's die goed aangeven om welke gegevensverzameling het gaat,
- een volledig overzicht van de relatie tussen transakties en werkplekken, zodat het gebruik van gegevens geografisch is bepaald.
- mogelijkheden en beperkingen ten aanzien van de opslag van gegevensverzamelingen. Denk daarbij aan de bestaande situatie, politieke of organisatorische beslissingen en dergelijke.

Gebruikers mogen best kiezen voor onlogische oplossingen als ze maar tijdig het prijskaartje hebben gezien. Automatisering is een ondeugdelijk middel om bedrijfspolitieke problemen mee op te lossen! De kunst is problemen te laten waar ze horen.
Tenslotte nog iets over transaktie-ontwerp in distributieve omgevingen in de verschillende fasen van de systeemontwikkeling.

- Tijdens de informatieplanning of het vooronderzoek.

Er kan een globale Transaktie analyse worden uitgevoerd. Per vestiging kan dan een globaal aantal beeldschermuren worden berekend. Als de topologie voor de gegevensopslag bekend is kan topologie van het netwerk worden getekend. Als de gegevensopslag nog bepaald moet worden en onder andere afhankelijk is van het netwerk, kan hoogstens een aantal mogelijkheden worden getekend. Wil men details van het netwerk dan moet transaktie-ontwerp worden uitgevoerd en verloopt het netwerkontwerp zoals tijdens het logisch ontwerp.

- Tijdens het logisch ontwerp.

Nu wordt het transaktie-ontwerp uitgevoerd en zijn na de logische Transaktie analyse de te verwachten technische resultaten bekend. Per lokatie is de situatie net zover bekend als in niet-

distributieve omgevngen. In hoeverre er iets gedaan kan worden aan de topologie van het netwerk, hangt af van de keuze ten aanzien van de distributie van de gegevens. Er moeten immers decentrale transaktieschema`s worden opgesteld.
- Tijdens het technisch ontwerp.
De technische Transaktie analyse met als basis centrale en decentrale transaktieschema`s levert nu alle gegevens voor het netwerkontwerp, omdat de kommunikatie tussen applikaties en de gegevens die worden uitgewisseld bekend zijn.

Hoofdstuk 43
Dialoogsimulatie

43.1 Dialoogsimulatie als methode

De dialoog is de kommunikatie met de computer via het beeldscherm: iedere druk op de ENTER-toets start een interaktie met de computer en na afloop van de responsetijd reageert de computer met een aantal tekens op het scherm. Wat er tijdens de responsetijd gebeurt is het resultaat van vele maanden werk van systeemontwerpers, programmeurs, database-ontwerpers en vele anderen. De gebruiker ziet alleen wat er op het scherm verschijnt. Hij voert een dialoog met het informatieverwerkende systeem. Alleen die dialoog wordt gesimuleerd. Er worden geen programma's ontwikkeld of bestanden ontworpen. Er wordt een prototype van de dialoog ontworpen, waar de gebruiker mee werkt. Evenals bij protoyping, hoeft dat niet te betekenen dat alle details gesimuleerd worden. In de praktijk zal het beeldscherm van de simulator er vaak anders uitzien dan het uiteindelijke beeldscherm van de gebruikers. Dialoogsimulatie is nu eenmaal geen terminalemulatie. Dialoogsimulatie is een methode om

- gebruikers inzicht te geven in de toekomstige situatie,
- gebruikers mede-ontwerper te laten zijn,
- een vaste vorm te hebben voor de kommunikatie tussen informatie-analisten en gebruikers.

Dat de resultaten van dialoogsimulatie ook bruikbaar zijn voor de

ontwerpers en in welke mate, is van secundair belang.
We zullen nu de methode in stappen beschrijven.
1. Kontroleren of de gebruikers zijn voorbereid op dialoogsimulatie. Zo niet, dan los van het projekt een workshop organiseren. (15)
2. De dialoogsimulator "thuis laten".
3. Met de gebruikers transaktieschema's maken.
4. Zoveel mogelijk attributen verzamelen van de transakties.
5. Op basis van de transaktieschema's startontwerpen maken voor schermlay-outs en transaktiedefinities.
6. Minstens een alternatief ontwerpen voor de dialoog.
7. Gebruikers selekteren of laten selekteren.
8. Het startontwerp demonstreren.
9. Het startontwerp evalueren en eventueel aanpassen.
10. De simulator in een praktijkomgeving plaatsen, zodat de gehele transaktie kompleet met aan- en uitloop, gebruik van brondokumenten en handmatige procedures, door gebruikers uitgevoerd kan worden.
11. De gebruiker minstens 10 tot 20 keer iedere transaktie laten uitvoeren.
12. Het noteren van enkele tijden.
13. Het ontwerp evalueren en eventueel aanpassen.
14. Het definitieve transaktieschema maken.
15. De definitieve transaktieschema's en schermlay-outs aan de gebruikers overhandigen.
16. De transaktieschema's overhandigen aan de transaktie-analist voor Transaktie analyse en aan de systeemontwerpers voor vergelijking met hun funktie- en gegevensmodellen.
17. De schermlay-outs en transaktiedefinities overhandigen aan systeemontwerpers.
18. De ontwerpfase voor de gebruikers expliciet afsluiten.

We zullen nu enkele van deze stappen toelichten.
Stap 1, 2 en 3 blijken voor veel informatie-analisten moeilijk te zijn. Hun liefde voor PC's is vaak zo groot, dat ze transaktieschema's snel vergeten. Het ligt zo voor de hand om meteen met de gebruiker achter de simulator te gaan zitten en schermlay-outs te gaan ontwerpen. Het ontwerpen van printlijsten blijkt velen nog in het batch-bloed te zitten. Maar een transaktie is iets anders dan een verzameling schermlay-outs. Een transaktie-ontwerp zal zeker resulteren in schermlay-outs, maar de volgorde is heel anders. Hoe relatief snel schermen ook te ontwerpen zijn op een simulator, het blijft vervelend om een hele opzet weer overhoop te moeten halen omdat bijvoorbeeld het werken met menuschermen toch niet zo handig blijkt. Nee, eerst alle transaktieschema's maken zodat een totaalbeeld ontstaat en dan pas overwegen wat een goede dialoogvorm is, is een veel betere volgorde. Dat betekent

dat er voorlopig nog geen dialoogsimulator nodig is: eerst worden samen met de gebruiker transaktieschema's gemaakt. Gebruikers die goed zijn voorbereid (15) weten wat transaktieschema's zijn.
Stap 4. Hier worden de attributen bedoeld die in de paragraaf Transaktie als entiteitstype zijn genoemd. Een deel daarvan is niet nodig bij dialoogsimulatie en hoeft dus niet nu verzameld te worden, hoewel het geen kwaad kan.
Stap 5 betekent dat er nu een totaal beeld is ontstaan, zodat ook de relatie tussen transakties en het aantal keren dat er gewisseld wordt tussen transakties, in aanmerking kunnen worden genomen bij het ontwerp.
Stap 6 is een heel belangrijke stap, waarvoor men zich meestal geen tijd zal gunnen. In de eerste plaats is er vakmanschap voor nodig om een andere dialoogvorm toe te passen, in de tweede plaats zit men heel gauw vast aan een eenmaal gekozen weg. Gebrek aan tijd is een handig argument en altijd aannemelijk te maken. Twee ontwerpen maken is om een aantal redenen gewenst. De eerste reden is dat een gebruiker die maar een ontwerp krijgt te zien, geen keuze mogelijkheid heeft en geneigd is aan te nemen dat dit ontwerp de enige mogelijke oplossing is. De tweede reden is dat het zien van alternatieven de gebruiker motiveert de beste oplossing te helpen zoeken en dat is die waar de gebruiker qua procedure tevreden over is en die de informatie-analist mogelijk acht. Op die manier worden betere oplossingen tijdens de akseptatietest aan het eind van het projekt voorkomen! Een situatie waarin een of meer alternatieven noodzakelijk zijn, is die waarin de wensen van de gebruikers leiden tot een ontwerp waarvan de informatie-analist van te voren weet dat het geen optimale oplossing biedt. Soms zijn gebruikers namelijk geneigd te komen met een ontwerp dat zoveel mogelijk overeenkomt met de handmatige situatie, maar de kombinatie beeldscherm-computer niet optimaal benut. Een vakman heeft de diverse dialoogvormen met hun plussen en minnen paraat (16), kent de mogelijkheden van besturing van het beeldscherm door de applikatie, kan lezen en schrijven met funktietoetsen en kent alle ergonomische aspekten van het uiteindelijke beeldscherm van de gebruiker. Wanneer de computerleverancier nog gekozen moet worden, valt hier uiteraard niet het onderste uit de kan te halen, maar een toetsenbord met funktietoetsen zit aan elk beeldscherm. In de paragraaf Transaktie-ontwerp in diverse omgevingen is al aangegeven dat het iteratieve ontwerpproces vanuit het technisch ontwerp, waar de computerleverancier in ieder geval bekend is, aanpassingen nog altijd mogelijk maakt.
Stap 7 zal in een groot bedrijf met een paar honderd eindgebruikers anders verlopen dan in een bedrijf waar slechts enkelen van de transakties gebruik zullen maken. In het laatste geval kan iedere gebruiker aan de simulatie deelnemen, in de eerste situa-

tie moeten de gebruikersafdelingen zelf uitmaken wat ze willen. In stap 8 zit de informatie-analist achter het beeldscherm. Hij bedient de simulator en vertelt hoe de transakties in elkaar zitten. Wanneer het doel was een volledig zelfverklarende dialoog te ontwerpen, vervallen stap 8 en 9.
Stap 9 is bedoeld om het kommentaar dat al tijdens de demonstratie is geleverd, te verwerken, schermlay-outs aan te passen, dialoogvormen te wijzigen etc. Wanneer het om wat details gaat, kunnen we meteen doorgaan met stap 10.
Stap 10 is het moment waar alles om draait. Hier zit de gebruiker in zijn nieuwe werkomgeving en hier worden de sociale aspekten uit de algemene vaagheid gehaald en konkreet en bespreekbaar gemaakt. Nu kan de gebruiker beoordelen of het werken met een beeldscherm past in zijn werkzaamheden en hoe de aansluiting is met de dokumenten waarmee hij werkt. Het zou toch werkelijk niet de eerste keer zijn dat een gebruiker prijzen moet wijzigen in artikelregels, waarvan de volgorde op het brondokument totaal anders is dan die op de schermen!
Stap 11. Wanneer het gaat om een groep van 3 tot 5 gebruikers hoeft natuurlijk niet iedereen 10 transakties uit te voeren, maar een of meer personen zeker wel.
Stap 12 is belangrijk voor Transaktie analyse. Zaken als aan- en uitloop, denk- en wachttijden zijn voor de transaktie-analist vaak moeilijk te schatten. Daarom is het verstandig de daarvoor benodigde tijd te noteren bij die handeling op het transaktieschema. In dit verband wijzen wij er nogmaals op dat de omgeving zo echt mogelijk moet zijn. Als het gaat om een kombinatie van telefoneren en werken met het beeldscherm, laat dan via een ander toestel een van de andere gebruikers opbellen. Noteer in ieder geval de totale transaktietijd.
Stap 13 hoeft natuurlijk niet meteen te worden uitgevoerd, als er nog verscheidene sessies volgen. Het hangt af van de vorm van het kommentaar. Bij een veld meer of minder op diverse schermen of een andere schermlay-out kan men beter gewachten tot alle sessies achter de rug zijn. Dan wordt in een keer alle kommentaar verwerkt en de uiteindelijke versie nog een keer gesimuleerd in een sessie waarin van alle groepen een persoon als vertegenwoordiger aanwezig is. Als tijdens de eerste sessie al zoveel kommentaar komt, dat de uiteindelijke oplossing zoveel zal afwijken van de oorspronkelijke versie dat niemand zich die goed kan voorstellen, is het beter meteen na de eerste sessie een nieuw ontwerp te maken en te simuleren met dezelfde groep. Is de groep gebruikers zo klein dat er maar een sessie nodig is kan de analist natuurlijk per geval beslissen of hij het kommentaar tijdens de sessie verwerkt of achteraf en een afspraak maakt voor een nieuwe sessie.
Stap 14 wordt uitgevoerd wanneer de uiteindelijke gesimuleerde

transaktie afwijkt van het oorspronkelijke transaktieschema.
In stap 15 krijgen de gebruikers dokumenten in handen die ze begrijpen en weer gebruiken tijdens de overdracht van het systeem.
Stap 16 en 17 zijn behandeld in de paragraaf Transaktie-ontwerp in verschillende omgevingen. Op deze manier van werken zijn natuurlijk varianten mogelijk, maar het doel en het resultaat moeten hetzelfde blijven: de gebruiker volgens een vaste methode betrekken bij het logisch ontwerp. Er zou dus niets op tegen zijn wanneer gebruikers zelf de stappen 2, 3, 5, 8, 9, 10 en 11 uitvoeren. Als het nodig is doet de informatie-analisten dan nog stap 6 en tijdens stap 10 geeft hij kommentaar vanuit zijn vakmanschap op het gebied van dialoogontwerp. In ieder geval zorgt hij ervoor dat stap 12 tot 17 korrekt worden uitgevoerd.
Stap 18 is van belang voor de voortgang van het projekt. Veel projekten lopen vertraging op, omdat gebruikers zelfs tijdens de bouwfase nog met nieuwe eisen komen. Wanneer de gebruikers goed zijn voorbereid, weten ze dat, ze in feite hun systeem besteld hebben als de dialoogsimulatie is afgerond. Alles wat daarna nog aan wensen en eisen naar voren komt, wordt genoteerd nog verwerkt in release 2 van de applikatie. Hoe meer gebruikers betrokken zijn geweest bij de simulatie, hoe kleiner de kans is dat er nog onmisbare zaken naar voren komen. Stap 18 is het schrijven van een interne mededeling aan alle betrokken gebruikers, dat alle kommentaar is verwerkt in het ontwerp en dat het systeem dienovereenkomstig gebouwd zal worden. Problemen tijdens het technisch ontwerp die gevolgen hebben voor de gebruikers, zullen betrokken worden bij de aanpassingen, zo nodig met behulp van de simulatie.
In de meeste gevallen zal het verstandig zijn, stap 18 pas uit te voeren tegen het eind van het logisch ontwerp.
Tenslotte kunnen we ons nog afvragen of alle transakties gesimuleerd moeten worden. Laten we in verband daarmee een paar belangrijke stappen in het transaktie-ontwerp nog even op een rijtje zetten.

- Transaktieschema's dienen voor de gebruikers als akseptatiedokument. Daarnaast zijn ze de basis voor Transaktie analyse.
- Dialoogsimulatie dient om de gebruiker inzicht te geven in zijn toekomstige omgeving.
- Transaktie analyse dient om cijfermateriaal te verzamelen over transakties. Transakties die eenmaal per dag of per week worden uitgevoerd zijn in't algemeen wat dit betreft niet interessant.

Een en ander betekent dat er op verschillende momenten beslissingen mogelijk zijn over het aantal transakties, waarvoor bovengenoemde drie stappen worden uitgevoerd. Een gemotiveerde informatie-analist doet er liever vijf teveel dan een te weinig, een analist die het toch al niet zo nodig vindt en bovendien in tijdnood zit, laat al gauw die transakties weg die maar enigzins op

een reeds behandelde transaktie lijken.
Het principe is dus: transaktieschema's en dialoogsimulatie voor alle transakties, behalve voor die, welke als twee druppels water op andere lijken.
Voor dialoogsimulatie geldt in sterke mate wat voor vele methoden op gaat: het vakmanschap van de informatie-analist is onmisbaar. Een informatie-analist die een presentatie over dialoogsimulatie voor gebruikers niet ziet zitten, kan uiteindelijk in dezelfde situatie terecht komen als zonder die methode: ontevreden gebruikers, die blijven ageren tegen het systeem en zijn ontwerpers. Daarom zullen we in de volgende paragraaf een aantal punten bespreken die fout kunnen gaan in de aanpak van de informatie-analist.

43.2 Problemen

Dialoogsimulatie is een methode om de materiedeskundigheid van de gebruiker te koppelen aan het automatiseringsvakmanschap van de informatie-analist. De gebruiker zal nooit een automatiseringsspecialist worden, maar hij moet wel werken met een beeldscherm. Hij mag best weten dat er maar 24 regels van 80 tekens op een scherm gaan en wat funktietoetsen zijn. In (15) worden dat soort zaken aan de gebruiker duidelijk gemaakt. Daar leert hij ook wat dialoogsimulatie inhoudt. In deze paragraaf gaat het niet om de problemen van de gebruikers, maar om de problemen die informatie-analisten soms hebben met het uitvoeren van de methode dialoogsimulatie.
- De voorbereiding van de gebruikers. Het is de taak van de informatie-analist de gebruiker voor te bereiden op samenwerking die tot stand moet komen. In (15) wordt in algemene termen gesproken over projektaanpak. De informatie-analist moet de gebruiker informeren over de grote lijnen van het onderhavige projekt. Wat is er gebeurd? Waar zijn we nu? Wat wordt er verwacht? Informatie-analisten klagen vaak dat gebruikers te weinig tijd hebben. Juist in die gevallen blijkt vaak dat de informatie-analist nog nooit een presentatie heeft gehouden met foils, overheadprojektor en dokumentatie. Niet dat de overheadprojektor maatgevend is, maar als die gebruikt is, is er in ieder geval een presentatie gehouden. Een vergadering met wat diskussies over wat er allemaal moet gebeuren zet geen zoden aan de dijk. Een presentatie plus een methode plus de geschatte tijd per transakties levert beschikbare en gemotiveerde gebruikers.
- Het ontwerpen van de schermlay-outs. Sommige informatie-analisten zitten zo vast aan het standaardpatroon van het ontwerpen van in- en uitvoerdokumenten dat het verschil tussen dialoogsimulatie en schermontwerp niet helemaal tot hen doordringt. Transaktie-

ontwerp is voor hen hetzelfde als schermontwerp en een dialoogsimulator een middel om snel schermen te ontwerpen. De gebruikers doen niets anders dan de getoonde schermen aksepteren. Soms komt er dan een aantal onhandige procedures aan het licht, bijvoorbeeld tijdens de akseptatietest. Zo kan een overzichtelijke schermlay-out heel slecht aansluiten op de brondokumenten: het lijkt dan alsof er dialoogsimulatie is toegepast doordat de simulator gebruikt is, maar het verschil met vroeger is slechts dat de schermlay-outs niet op papier staan maar op een scherm. Er zijn dan nog geen transakties ontworpen samen met de gebruiker, er zijn geen transaktieschema's, er wordt geen Transaktie analyse uitgevoerd en de interaktieve toepassingen worden niet gekwantificeerd.

- Tegenzin tegen veranderingen. Het automatiseringsvak is op zichzelf al ingewikkeld genoeg. Iedereen heeft het liefst zijn eigen vertrouwde manier van werken. Automatiseringsspecialisten hebben het wel eens over de veranderingsanalyse in de gebruikerswereld. Daarbij gaat het dan over de gevolgen van invoering van de automatisering. Soms lijkt het erop dat een veranderingsanalyse binnen de automatisering minstens even hard nodig is. Dan gaat het om de verandering van het ontwerpen van gegevens, funkties en programma's naar het ontwerpen van transakties. Niet dat het eerste niet meer nodig zou zijn, maar het is niet het enige. Bij het ontwerpen van interaktieve toepassingen komen de schermlay-outs het laatst. De volgorde is: de huidige werkzaamheden in kaart brengen, de transakties ontwerpen, de dialoog ontwerpen en als sluitstuk is daarbij natuurlijk een schermlay-out nodig. In negen van de tien gevallen is de schermlay-out helaas het enige en het eerste wat er te vinden is in de systeemdokumentatie van de interaktieve toepassingen!

- Beperkingen van de simulator. De meeste informatie-analisten zijn automatiseerders pur sang: er zijn maar weinig bedrijven die gebruikers opleiden tot informatie-analisten. Dat is voor dialoogsimulatie meestal een handicap. De technische informatieanalisten denken ook tijdens het logisch ontwerp al vaak te veel in termen van programma's en bestanden. Wanneer in een bepaalde transaktie de gebruiker twee bedragen moet intypen en de computer de som moet displayen, terwijl de simulator een willekeurig bedrag displayd, is dat voor hen een onoverkomelijk probleem. De informatie-analist die uit het goede hout is gesneden zal dat effekt gebruiken om de gebruiker er op te wijzen dat het uiteindelijke programma dat voor de optelling en al het andere zorgt, nog gebouwd moet worden. De eenvoudigste gebruiker begrijpt dat heel goed en vindt het prima. Deze zaken zijn hem immers tijdens de voorbereiding (15) al duidelijk geworden.

Wanneer de simulator geen echte foutsituaties aan kan, is het al-

tijd mogelijk om een foutscherm aan een funktietoets te koppelen zodat het op elk moment kan worden opgeroepen. De echte foutsituatie die later door een programma wordt ontdekt, moet nu gesimuleerd worden met een druk op een funktietoets. Zo zijn er legio voorbeelden te noemen. De beperkingen van de simulator zijn op een gegeven moment bekend en de gemotiveerde informatie-analist wordt een uitstekende demonstrateur. Hij laat zich door de beperkingen van de simulator niet van zijn doel afbrengen: de gebruiker inschakelen als mede-ontwerper.

- Dialoogsimulatie is geen terminalemulatie. Het kan best zijn dat het toetsenbord van de dialoogsimulator verschilt van het uiteindelijke beeldscherm. In de praktijk is dat geen enkel probleem voor de gebruiker. De funkties zijn immers in principe gelijk en verschillen worden ongedaan gemaakt door de demonstrerende informatie-analist, zoals aangegeven in het vorige punt. Het verschil in tijd tussen logisch ontwerp en oplevering is zo groot, dat de gebruiker bij de akseptatietest dat andere toetsenbord al lang vergeten is. Met het transaktieschema en de schermlay-out in de hand, kontroleert hij de gebouwde transakties, die wat de dialoog betreft nog steeds moeten werken zoals destijds op papier is vastgelegd.

- Automatiseerders zien de simulator te vaak als gereedschap van en voor de automatiseerders. Eigenlijk zou de simulator meteen programma's en databases moeten afleveren! De data dictionary zou er meteen in moeten zitten om kontrole op naamgeving van schermen en velden mogelijk te maken. Soms worden deze zaken als argumenten gebruikt om de simulator niet te gebruiken. En dat terwijl de simulator slechts het middel is binnen de methode. Wanneer er gekozen is voor dialoogsimulatie als methode voor de samenwerking met de gebruikers, dan wordt vervolgens de best bruikbare simulator toegepast, inclusief wat naar het oordeel van de automatiseerder gebreken zijn. Doorslaggevend is de keuze voor de methode en de mening van de gebruiker over de simulator.

Hiermee is een aantal problemen genoemd uit de praktijk. Het zijn geen onoplosbare problemen. Het probleem is de bril, die we als automatiseerders allemaal op hebben. Als we eens een andere bril proberen, zal blijken dat er veel veranderd. Vooral de bril van de gebruiker, voor wie we uiteindelijk aan het werk zijn, blijkt verhelderend te zijn!

43.3 De dialoogsimulator

De dialoogsimulator bestaat niet. Wel bestaan er allerlei hulpmiddelen om dialoogsimulatie uit te voeren. In principe komt een dialoogsimulator neer op een beeldscherm gekoppeld aan een programma en een gegevensbank. De informatie-analist moet alleen de

schermlay-outs vastleggen. Het programma zorgt voor kommunikatie tussen het beeldscherm en de gegevensbank. Een dialoogsimulator kan een funktie van een mainframe zijn, het kan ook een pakket op een microcomputer zijn (22). Het voordeel van een funktie op een mainframe is dat het beeldscherm van de simulator meestal gelijk is aan dat van de uiteindelijke toepassing. Zoals in de vorige paragraaf is aangegeven is dat geen noodzaak, maar wel een voordeel. Een nog groter voordeel voor de micro is, dat die meestal op elke werkplek kan worden neergezet. In decentrale bedrijven waar al micro's in gebruik zijn, hoeft de informatie-analist alleen maar floppies mee te nemen. (Natuurlijk alleen als het bedrijfsbeleid voorschreef, dat er maar een soort micro mocht worden aangeschaft!)

Naast het vastleggen van schermlay-outs moet natuurlijk ook de relatie tussen de schermen kunnen worden beschreven: een transaktie zal immers vaak uit verscheidene schermen bestaan. Het invoeren van de relatie tussen de schermen noemen we de transaktiedefinitie. Soms worden niet alle schermen van transakties gesimuleerd. Dat betekent dat de transaktiedefinitie niet altijd gelijk zal zijn aan de uiteindelijke dialoogstruktuur, waarin alles exakt en voor 100 procent vastligt. De transaktiedefinitie werd in het vorige hoofdstuk dan ook steeds genoemd als basis voor het dialoogstruktuurdiagram.

In de praktijk blijkt het voldoende te zijn, wanneer op drie manier kan worden aangegeven wanneer een scherm moet verschijnen:

- nadat op het voorafgaande scherm het laatste veld is ingetypt.
- op basis van de inhoud van het laatste ingetypte veld van het voorgaande scherm. Dit maakt een boomstruktuur en dus bijvoorbeeld menu-simulatie mogelijk.
- als gevolg van het indrukken van een funktietoets. Dit maakt het mogelijk op elk willekeurig moment een bepaald scherm te laten verschijnen. Dit is een belangrijk hulpmiddel in de handen van een goede demonstrateur. Daarnaast kan dit mechanisme uiteraard worden gebruikt om funktietoetsen te gebruiken zoals de uiteindelijk in de applikatie ook zullen worden toegepast.

Dialoogsimulatie bestaat dus uit drie stappen:

1. het ontwerpen van schermen.
2. het vastleggen van de transaktiedefinitie.
3. het simuleren van een transaktie.

Het ontwerpen van schermen doen we in twee stappen.

1A. Het maken van de schermlayout. Alle tekst op het scherm wordt op de juiste plaats neergezet. Met funktietoetsen kan nog een aantal "screenpainting"-funkties worden uitgevoerd. Alle invoer- en displayvelden worden aangegeven met een besturingsteken, bijvoorbeeld een open- en een sluithaak.

1B. De velddefinitie. In deze stap moet van elk veld worden aange-

geven of het een intoetsveld of een displayveld is. Van displayvelden moet worden aangegeven welk gegeven uit de gegevensbank gedisplayed moet worden en op welk moment binnen de dialoog. Een bijzonder soort intoetsvelden zijn de KEUZE-velden, op basis waarvan binnen een boomstruktuur gesprongen wordt.
In stap 2 wordt per transaktie een tabel ingevuld met de volgende kolommen: schermnaam, opgeroepen door scherm, opgeroepen door een KEUZE-veld met de inhoud, opgeroepen door funktietoetsnummer.
De eerste twee kolommen moeten altijd zijn ingevuld behalve voor het eerste scherm. De andere kolommen kunnen zijn ingevuld.
Het eerste scherm wordt niet opgeroepen door een ander scherm, maar door het programma. Dat programma zorgt er tevens voor dat het eerste scherm weer verschijnt nadat van het laatste scherm het laatste veld is ingetoetst. Overigens mogen in de kolom schermnaam behalve namen van schermen ook namen van transakties genoemd worden. Binnen een transaktie kan dus een transaktie worden opgeroepen, maar steeds geldt dat een transaktie een zichzelf repeterende serie schermen is. Met een funktietoets kan naar het oproepende niveau teruggesprongen worden. Afgezien van dit detail is een transaktie op de dialoogsimulatie een getrouwe weergave van wat er op het transaktieschema is aangegeven en in het algemeen vele malen per dag door een gebruiker aan zijn beeldscherm zal worden uitgevoerd.
Nadat een of meer transakties gedefinieerd zijn kan er al gesimuleerd worden. Dus wanneer een transaktie slechts uit een scherm bestaat komt de hele voorbereiding neer op het maken van de schermlay-out, de definitie van de velden en het invullen van een regel in de transaktiédefinitie, zie Fig. 43.1.
Wanneer stap 3, het simuleren wordt gestart verschijnt het scherm met de cursor achter KLANTNUMMER: Wanneer een nummer is ingetypt verschijnt er een willekeurige firmanaam achter NAAM:, een adres achter ADRES:, een bedrag achter BEDRAG: en een bedrag achter PRIJS P.ST:. De cursor staat achter PRIJS P.ST.:, omdat dit veld gewijzigd moest kunnen worden. Nadat de RETURN-toets wordt ingedrukt gaat de cursor naar het volgende invoerveld, veldnummer 6. Wanneer na het invullen daarvan de RETURN-toets wordt ingedrukt, is dit scherm afgehandeld, en volgens de transaktiedefinitie tevens de transaktie. Dat betekent dat het eerste scherm van de transaktie weer getoond wordt, in dit geval S1.
Het principe is eenvoudig en iedere informatie-analist moet de simulator in een halve dag kunnen leren bedienen. Dat is wat anders dan welke vorm van prototyping dan ook. Prototyping is dan ook niet te vergelijken met dialoogsimulatie, hoewel het verschil voor de gebruiker meestal erg klein is.
Na enige jaren praktijkervaring met een dergelijk gereedschap, blijkt dat er meer nodig is, dan de bovengeschetste mogelijkhe-

```
Klantnummer:[             ]

NAAM       :[                          ]

ADRES      :[                          ]

BEDRAG     :[             ]

PRIJS P.ST.:[             ]          AKKOORD J/N: [   ]
```

Het scherm S1

Veldnr. 1: I(NVOER), responsetijd: 2 sec.

Veldnr. 2: D(ISPLAY), FIRMANAAM, na veldnr. 1

Veldnr. 3: D, ADRES, na veldnr. 1

Veldnr. 4: D, BEDRAG, na veldnr. 1

Veldnr. 5: D, te wijzigen, BEDRAG, na veld 1

Veldnr. 6: I, responsetijd: 1 sec.

De velddefinitie van S1

TRANSAKTIE: T1

NAAM	OPROEPER	KEUZE	F-TOETS
S1			

De transaktiedefinitie van T1

Fig. 43.1 Een voorbeeld op een dialoogsimulator

den: schermen moeten gemakkelijk te wijzigen zijn, met of zonder aanpassing van de velddefinitie, er moet met subschermen gewerkt kunnen worden om start/stopschermen na te bootsen, de gegevensbank moet aan te passen zijn, overzichten van gemaakte schermen en transakties moeten kunnen worden getoond, bepaalde gegevens moeten geprint kunnen worden enz. Een voorbeeld van een dergelijke simulator (22) wordt gebruikt in (15) en (16). Op een microcomputer is het een uitstekend gereedschap voor de informatieanalist die nu volledig onafhankelijk van de beschikbaarheid van beeldschermen en lijnen, op elke willekeurige plaats met de gebruikers aan het werk kan.
De uitvoering op een micro zou uit twee floppies kunnen bestaan: een systeemfloppy met het programma en de standaard gegevensbank en een werkfloppy die na initialisering een kopie van de gegevensbank bevat, die kan worden gewijzigd, en alle schermen en transakties. Per projekt kunnen we een werkfloppy nemen.
In het hoofdstuk Transakties is de relatie tussen methoden als dialoogsimulatie en de andere systeemontwikkelingsmethoden uitgebreid behandeld. De tastbare resultaten van dialoogsimulatie zijn de schermlay-outs, transaktiedefinities en transaktieschema's.De eerste twee bevinden zich na afloop van de simulatie op de werkfloppy van de simulator. Er bestaat een aantal mogelijkheden voor de koppeling met andere projektaktiviteiten.
De eenvoudigste koppeling bestaat uit het printen van schermlay-outs en transaktiedefinities, wat betekent, dat de schermen later opnieuw moeten worden ingetypt. De transaktiedefinitie dienst als basis voor de dialoogstruktuur en dus is een dokument voldoende. Een andere koppeling is die via floppies. Op het uiteindelijke systeem worden de schermlay-outs van de werkfloppies overgezet naar de schermbibliotheek. In het algemeen is daar een konversie voor nodig, maar meestal gaat dat met een simpel vertaalprogramma. Er moet wel onderzocht worden of de schermbibliotheek wel schermlay-outs aksepteert die op deze manier worden aangeleverd. Een derde mogelijkheid is gebruik maken van een d.c.-verbinding om de schermen over te sturen. Ook in dit geval moet bekeken worden hoe de schermen geaksepteerd worden door de ontvangende computer en of ze in de schermbibliotheek kunnen worden opgenomen.
In deze paragraaf is het gereedschap in grote lijnen behandeld. Als er gekozen wordt voor de methode dialoogsimulatie, moet vervolgens gezocht worden naar een bruikbare simulator. Daarbij moet gelet worden op reeds aanwezige hardware en software, portabiliteit, koppelingsmogelijkheden met andere methoden, kosten bij eigen ontwikkeling en bij aankoop van een simulator (22). Men kan besluiten om een bestaande mappingsupport-funktie te gebruiken of een dialoogsimulator op de bestaande konfiguratie te ontwikkelen met gebruikmaking van reeds beschikbare funkties of een dialoog-

simulator aan te schaffen en te (laten) konverteren naar de bestaande konfiguratie of een dialoogsimulator te kopen die draait op een beschikbare of aan te schaffen micro.
De dialoogsimulator zou voorzien kunnen worden van een aantal funkties waarmee tijden gemeten worden die bij Transaktie analyse geschat moeten worden. In dat geval zou men kunnen spreken van een transaktiesimulator.

43.4 Responsetijden

De problemen rond responsetijden zijn algemeen bekend. Iedereen kent de kwaal, maar niemand kan even de oorzaak aanwijzen. Een responsetijd is opgebouwd uit een aantal ingewikkelde elementen die elk meestal een vakgebied betreffen. Om er een paar te noemen: het transport door het netwerk, de verwerking door de applikatie, de T.P.-monitor en het operatingsysteem. In de praktijk zijn de systeemontwerpers kennelijk steeds weer zo verdiept in het logische aspekt van hun ontwerp, dat niemand zich echt zorgen maakt over de responsetijd. Er zijn zoveel andere zaken dan de applikatie, die de responsetijd beinvloeden, dat de ontwerpers er maar het beste van hopen. Veel evaluaties van interaktieve toepassingen bewijzen dat. Aan welke kronkels in het ontwerp de gebruiker ook went, althans volgens de ontwerpers, nooit aan lange responsetijden. De ergenis daarover neemt alleen maar toe. Nog vreemder wordt de zaak, als blijkt dat voor de meest voorkomende, meest irritante en kostbaarste fout in de automatisering niet eens eisen bestaan. De opmerking in de systeemspecifikatie dat de gemiddelde responsetijd korter moet zijn dan twee seconden, is geen eis. Wat is een gemiddelde responsetijd? Het gemiddelde van de responsetijden van alle interakties in alle toepassingen van het projekt? Over die eis hoeft geen enkele ontwerper zich zorgen te maken! Hoe zou hij trouwens met die zorgen om moeten gaan? In de breedte gezien zijn er vaak nog andere systeemontwerpers bij het projekt betrokken en in de diepte gezien zijn er altijd allerlei andere specialisten verantwoordelijk voor hun aandeel in de responsetijd.
Met andere woorden, er bestaan in de praktijk geen eisen waar iemand konkreet iets mee kan doen en er kan dus niemand worden aangekeken op het resultaat. De informatie-analist is verantwoordelijk voor de eisen, ontwerpers en bouwers moeten zorgen voor de realisering. Dat kan overigens ook betekenen dat de ontwerpers zorgen dat de eisen worden aangepast voor de bouw begint. Aangezien de eisen door de gebruikers zijn gesteld, kan aanpassing van de eisen alleen in overleg met hen.
In het kader van dialoogsimulatie gaat het alleen om het stellen van de eisen. Zoals reeds is opgemerkt heeft een opmerking over

gemiddelde responsetijden geen zin. Bij dialoogsimulatie komen echter alle interakties voor het voetlicht. Alle transakties worden in kaart gebracht op een transaktieschema en voorzien van een naam. Vervolgens worden de transakties gesimuleerd met de gebruiker. Op de dialoogsimulator moet de responsetijd per interaktie instelbaar zijn en een ervaren informatie-analist kan schatten of een verwerkingsproces langer dan 10 seconden zal duren of dat het binnen 1 of 2 seconden is gebeurd. In ieder geval is het van belang om alle responsetijden in het begin in te stellen op 5 seconden of langer. De gebruiker ervaart dan heel konkreet hoe lang 5 of meer seconden duren. Alleen van de interakties die de gebruiker echt niet akseptabel vindt, worden de tijden korter ingesteld. Gebruikers die gewoon vinden dat alles flitsend moet gaan op een beeldscherm, moeten gekonfronteerd worden met de beperkingen van het bestaande systeem of met de prijs van het systeem dat past bij hun eisen. In het algemeen zijn gebruikers, die ervaren dat ze konkreet betrokken worden bij het ontwerp, heel redelijk en begrijpen ze dat niemand gediend is met het stellen van onredelijke eisen. Het behoort tot het vakmanschap van de informatie-analist om aan te voelen of de gebruiker terecht hoge eisen stelt of dat zijn reaktie het gevolg is van andere problemen, bijvoorbeeld ten aanzien van de sociale aspekten van de automatisering. De instelling van responsetijden op de simulator is van groot belang in de strijd tegen te lange responsetijden. Het is verbazingwekkend hoe informatie-analisten soms hun schouders ophalen over dit gebeuren en dezelfde beweging maken als een jaar later extreme responsetijden aan de gebruikers verkocht moeten worden. In diverse delen van dit boek behandelen we de technische aspekten van responsetijden, maar alles begint bij konkrete eisen. Waar die ontbreken, zal nooit iets veranderen. Hoe snel computersystemen in de toekomst ook mogen worden, we zullen die kapaciteit inplannen voor ze beschikbaar is.

Dialoogsimulatie biedt de mogelijkheid om met de gebruiker de gespecificeerde eisen vast te stellen. Op het transaktieschema wordt bij iedere pijl naar links de gewenste responsetijd gezet. Wanneer de transaktieschema's bij de systeemontwerpers terecht zijn gekomen, moeten ze daar op een aantal aspekten worden bekeken, zoals in hoofdstuk Transakties is aangegeven. Een van die aspekten is de responsetijden. Hoewel we ons dan nog maar in de fase logisch ontwerp bevinden, moeten systeemontwerpers al in staat zijn te attenderen op eisen die in ieder geval niet haalbaar zijn. Dat geeft dan aanleiding tot de eerste iteratie in het ontwerpproces. Tijdens het technisch ontwerp worden de eisen opnieuw bekeken. Dan is immers per interaktie precies bekend wat de verwerking door de conputer inhoudt.

Soms komen informatie-analisten met het bezwaar dat ze het teveel

werk vinden om iedere responsetijd met de gebruiker te evalueren tijdens dialoogsimulatie. Als we ons realiseren dat iedere interaktie waarschijnlijk vele mandagen kost aan ontwerp, bouw en dokumentatie, is dit een vreemd bezwaar. Zeker als we nog in aanmerking nemen, dat er eigenlijk geen weg terug is: eenmaal gebouwde systemen blijven jaren in gebruik.
Beheersing van responsetijden zal in ieder geval niet gerealiseerd worden, zolang men het stellen van konkrete eisen overbodig acht.

43.5 Dialoogsimulatie als hulpmiddel voor de analyse

Dialoogsimulatie is bedoeld als ontwerpmethode en hoort daarom thuis in de fase logisch ontwerp. In de analysefase wordt de bestaande situatie in kaart gebracht. Zolang het gaat om keurig beschreven processen en procedures, is dat geen probleem, maar het wordt moeilijker bij procedures in de hoofden van medewerkers. Via interviews en beschrijvingen moet dan getracht worden te analyseren wat er precies gebeurt. Gezien het gemak waarmee op een dialoogsimulator schermen kunnen worden ontworpen en reakties van een systeem kunnen worden gesimuleerd, kan dialoogsimulatie in sommige gevallen worden toegepast, om de gebruiker via het beeldscherm zijn manier van werken in kaart te helpen brengen. Nogmaals, dit heeft niets te maken met ontwerp van de nieuwe situatie.
Men kan daarbij denken aan het maken van overzichten van entiteitstypes, de erbij behorende attributen, het aantal tekens, het formaat en dergelijke. Wanneer deze gegevens bijvoorbeeld per dokument of per procedure verzameld worden, kunnen de schermlayouts misschien dienen als basis voor het startontwerp. In ieder geval is dan de terminologie al afgestemd op de gebruiker.

Hoofdstuk 44
Transaktie analyse

44.1 Transaktie analyse

Transaktie analyse is een methode om eisen van de gebruikers te kwantificeren en om te rekenen naar konsekwenties voor gebruikers, computersysteem en netwerkontwerp.
We zullen eerst de belangrijkste termen uit deze definitie bespreken
- Een methode. Transaktie analyse is een manier van werken die uit de volgende stappen bestaat:
1. het maken van detailschema's op basis van transaktieschema's. Daarmee is een rekenmodel van de transaktie ontstaan.
2. het invoeren van het detailschema in een rekenprogramma.
3. het verwerken van de resultaten van het rekenprogramma tot konklusies. Er is maar een rekenprogramma, dat altijd dezelfde resultaten oplevert, die niet altijd alle relevant zijn.

Daarmee is tevens de relatie aangegeven met transaktie-ontwerp en dialoogsimulatie, zoals die behandeld zijn in het hoofdstuk Transakties. Het transaktieschema wordt gebruikt als start voor Transaktie analyse, onafhankelijk van het feit of het met de gebruiker via dialoogsimulatie ontwikkeld is of niet. Transaktie analyse en dialoogsimulatie sluiten op elkaar aan en zijn met elkaar verweven tot transaktie-ontwerp, maar beide methoden zijn ook onafhankelijk van elkaar bruikbaar.

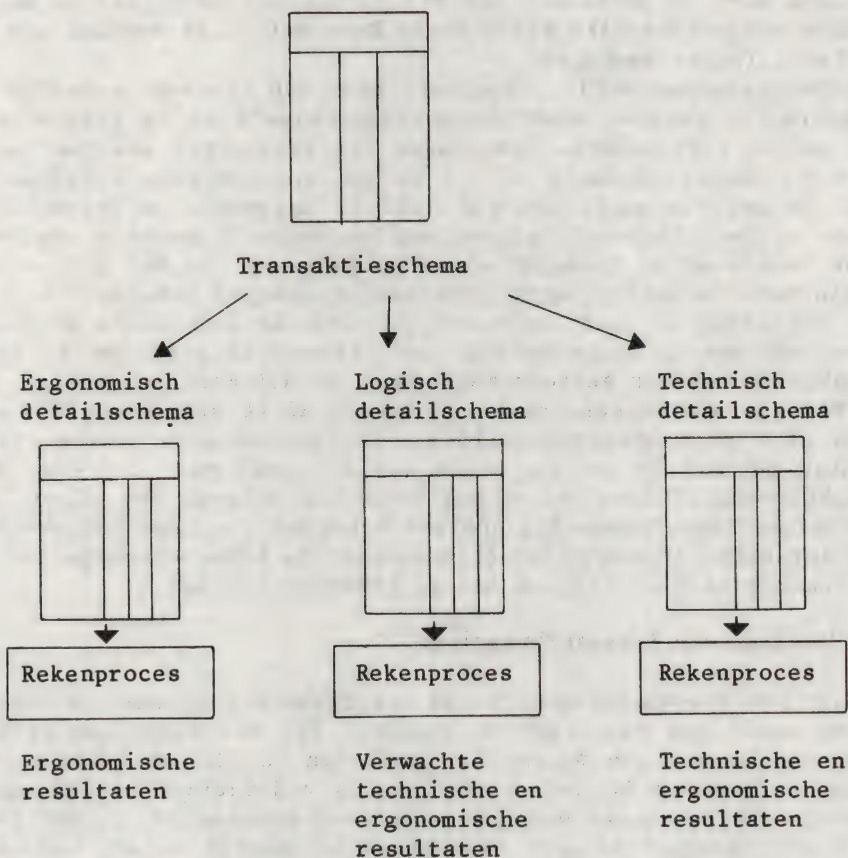
- Eisen van gebruikers. Het gaat daarbij om eisen die zijn vastgelegd op transaktieschema's. Op die schema's komen in principe geen cijfers voor. Ze bevatten een kwalitatieve omschrijving van de nieuwe procedure voor de gebruiker en de gewenste verwerking door de computer. Het transaktieschema is een dokument van de gebruiker en de informatie-analist. Of de wensen haalbaar zijn, wat performance, kosten, aanwezige hardware en software betreft, moet nog onderzocht worden.
- Kwantificeren. Het detailschema is het gekwantificeerde transaktieschema. Als er in het transaktieschema staat dat de gebruiker moet bladeren in schermen met artikelregels, dan staat in het detailschema hoe vaak dat per transaktie gebeurt. De gebruiker moet aangeven in hoeveel orderregels hij denkt te moeten zoeken. Als er op het transaktieschema staat dat de gebruiker een zoekargument moet intypen, staat in het detailschema hoeveel tekens er moeten worden ingetypt. Aangezien de meeste van deze gegevens door de gebruikers verstrekt moeten worden is het van belang dat informatie-analisten iets weten van Transaktie analyse, zodat ze de juiste cijfers kunnen verzamelen. We noemen dit kwantiteiten binnen transakties.
- Omrekenen naar konsekwenties. Het rekenprogramma levert voor iedere transaktie drie pagina's output plus het detailschema dat voor de resultaten heeft gezorgd. Er ontstaan twee soorten resultaten: ergonomische en technische. Een ergonomisch resultaat is bijvoorbeeld de tijd die een transaktie duurt. Wanneer bekend is hoeveel transakties er per dag uitgevoerd moeten worden kan dat worden omgerekend naar het aantal benodigde beeldschermen, beeldschermuren per gebruiker per dag, de doorlooptijd, de uitloop van werk naar de volgende dag, wachtrijen en dergelijke. De gebruiker moet gegevens verstrekken over het aantal transakties per dag, de verdeling tussen transakties, de vereiste doorlooptijd enzovoort. Dit noemen we kwantiteiten van transakties.
- Konsekwenties voor gebruikers. Het aantal uren per dag dat een gebruiker doorbrengt achter een beeldscherm is een belangrijk aspekt van zijn nieuwe taak. Dit soort gegevens is bijna nooit van te voren bekend. Bij een methode als Transaktie analyse ontstaan ze al tijdens het logisch ontwerp en maken ze deel uit van het ontwerpproces waarbij de gebruiker is betrokken. Op grond van deze gegevens moeten gebruikers 'nee' mogen zeggen tegen een ontwerp. Daarna zullen ze zeer gemotiveerd willen helpen zoeken naar alternatieven. Dan wordt het iteratieve ontwerpproces ook in de automatisering een konkreet gebeuren.
- Konsekwenties voor het computersysteem. Sommige resultaten van Transaktie analyse kunnen worden omgerekend naar kengetallen voor de "zwaarte" van transakties. Uiteindelijk zullen op een aantal beeldschermen transakties worden verwerkt. Een kleine computer

kan maar enkele beeldschermen met lichte transakties aan. Voor zwaardere transakties is een groter systeem nodig. Voor veel beeldschermen met hele zware transakties is een mainframe nodig. De termen licht en zwaar hebben geen enkele waarde. Sommige denken bij een zware transaktie aan een complex rekenprogramma, anderen aan veel I/O`s op een database. Maar hoe iedereen ook denkt, niemand doet het in cijfers. Transaktie analyse levert kengetallen op een transaktie te karakteriseren, qua systeembelasting.
In het algemeen is dat geen zaak voor de informatie-analisten. Daarom wordt Transaktie analyse meestal uitgevoerd door transaktie-analisten. De ergonomische resultaten zal hij doorgeven aan de informatie-analisten, die daar hun konklusies uit trekken en die bespreken met gebruikers. De technische resultaten komen terecht bij degenen die zich bezighouden met systeembeheer, performance onderzoek of aanschaf van een nieuwe computer.
- Konsekwenties voor het netwerkontwerp. Wanneer het beeldscherm van de gebruiker via een netwerk aan de computer wordt gekoppeld, is het belangrijk de hoeveelheid verkeer, in tekens per seconde, tussen beeldscherm en computer te bepalen. Hoe meer tekens er op een scherm staan en hoe vaker de gebruiker op de ENTER-toets drukt, hoe meer tekens er per seconde getransporteerd moeten worden. Daarbij speelt ook de intelligentie van het beeldscherm een rol. Op basis van de in het transaktieschema weergegeven dialoog, gekoppeld aan de kennis van het soort beeldscherm, maakt de transaktie-analist het detailschema. Het analyseprogramma levert resultaten over het verkeer door het netwerk. Wanneer verder bekend is op welke lokaties transakties worden uitgevoerd en op hoeveel beeldschermen, kunnen de netwerkontwerpers aan de slag. De konsekwenties voor computersysteem en netwerkontwerp lijken dus niet teruggekoppeld te worden naar de gebruiker. Toch is dat wel de bedoeling. Het netwerkontwerp zal echter meestal pas plaatsvinden tijdens het technisch ontwerp. Als daar zou blijken dat het netwerk te duur wordt, komt de terugkoppeling alsnog! Nu gebeurt dat in de praktijk nooit omdat, zeker wat het netwerk betreft, het technisch ontwerp geruisloos overgaat in de bouw. Daarna is er voor geen enkel bedrijf meer een weg terug. Blijkt het ontworpen netwerk te weinig kapaciteit te bezitten dan wordt het hoogstens uitgebreid, maar nooit afgebroken, en dat verklaart de gerechtvaardigde angst van veel managers voor steeds nieuwe budgetten voor onvoorziene tegenvallers. Konsekwente toepassing van Transaktie analyse levert al tijdens het technisch ontwerp een realistisch kostenplaatje: dan komt de terugkoppeling naar de gebruikers of hun management dus wel degelijk. Dan merkt de informatie-analist dat misschien toch weer, omdat gezocht moet worden naar transakties die een goedkoper netwerk mogelijk maken.

Misschien moet de gebruiker dan wel aardig wat water bij de wijn van zijn oorspronkelijke eisen doen. Maar dat is de normaal bij alle bedrijfsinvesteringen.
Transaktie analyse wordt uitgevoerd door een transaktie-analist. De informatie-analist maakt transaktieschema's en is geinteresseerd in de ergonomische resultaten van Transaktie analyse. Hij levert transaktieschema's af en krijgt ergonomische resultaten terug. Informatie-analisten die zelf de ergonomische Transaktie analyse willen uitvoeren, moeten het hoofdstuk Transaktie analyse in het deel voor de transaktie-analisten lezen. In dit deel voor de informatie-analisten wordt Transaktie analyse behandeld voorzover dat nodig is voor het maken van transaktieschema's en voor het omgaan met de ergonomische resultaten. In principe is het transaktieschema een gebruikersdokument en dat moet het ook blijven. Het kan echter geen kwaad de inhoud af te stemmen op wat er daarna door de transaktie-analisten mee gedaan moet worden. Als dat niet gebeurt is er nog niets aan de hand, maar dan komt de transaktie-analist wat vaker met bepaalde vragen. We zullen nu eerst de soorten Transaktie analyse behandelen en dan terugkomen op de interface tussen informatie-analist en transaktie-analist in de paragraaf Kwantiteiten binnen transakties. (44.3)

44.2 Vormen van Transaktie analyse

De start van Transaktie analyse is het transaktieschema. Op basis daarvan wordt het detailschema gemaakt. Bij het maken van beide dokumenten is het van belang het doel van de analyse in het oog te houden. Wanneer dat het verkrijgen is van inzicht in het aantal uren dat gebruikers bezig zijn met een transaktie, is het zeker in een Cx-omgeving niet belangrijk het aantal tekens van een schermlay-out mee te nemen in het detailschema. Het kan geen kwaad omdat het rekenproces altijd hetzelfde is, maar het is zonde van de tijd. Het voorkomen van tijdverspilling is een reden waar om we onderscheid maken in vormen van Transaktie analyse. De tweede reden is de beschikbaarheid van de gegevens. Tijdens het logisch ontwerp is de verwerking alleen in principe bekend. Hoe bestanden en databases er echt uitkomen te zien is dan nog niet bekend. Toch is het dan al van belang de eerste indikatie van bijvoorbeeld de responsetijden op papier te hebben. Dat soort indikaties kan immers de eerste start zijn voor een iteratie in het ontwerpproces.
Gezien de te bereiken resultaten zijn er drie vormen van Transaktie analyse onderscheiden. Zie Fig. 44.1.
- Ergonomische Transaktie analyse. Bij deze vorm gaat het om het verkrijgen van de ergonomische attributen van iedere transaktie, zoals beschreven in de paragraaf Transaktie als entiteitstype.



Transaktieschema:

- Globaal : b.v. tijdens vooronderzoek
- Voorlopig : geen dialoogsimulatie uitgevoerd
- Definitief: na dialoogsimulatie.

Fig. 44.1 Drie vormen van Transaktie analyse

Het transaktieschema moet nu zo nauwkeurig mogelijk de menselijke handelingen beschrijven. De verwerking door de computer wordt met een paar woorden gekarakteriseerd. Natuurlijk hoort de responsetijd tot de ergonomie, maar in een transaktie gaat het meestal om meer responsetijden. De eisen ten aanzien van die responsetijden staan in het transaktieschema vermeld. Responsetijden komen voor het voetlicht bij de twee andere vormen van Transaktie analyse. In het algemeen is het zo dat de responsetijd te verwaarlozen is ten opzichte van de totale transaktietijd. Daarom wordt bij de ergonomische analyse door de transaktie-analist de responsetijd op een redelijk geachte tijd ingesteld. Wanneer men ook de verwerking door de computer in de analyse wil betrekken, wordt een logische Transaktie analyse uitgevoerd.
De resultaten van Transaktie analyse zijn altijd drie pagina's met cijfers: het parameteroverzicht, de terminaltransaktietijd en de lijn- en responsetijdaspekten. Bij de ergonomische analyse gaat het om de pagina terminaltransaktietijd. Die pagina levert de tijd die een transaktie duurt en de samenstelling van die tijd. In de paragraaf Resultaten van Transaktie analyse en konklusies komen we uitgebreid op deze pagina terug.
- Logische Transaktie analyse. Logisch slaat op de logische verwerking. Tijdens het logisch ontwerp kan de verwerking door de computer alleen worden aangegeven in logische accessen tot gegevens. Aangezien de keuze van de eenheid waarin de verwerking wordt uitgedrukt vrij is, kan deze eenheid worden aangepast aan het soort systeem, de kennis van dat systeem tijdens logisch ontwerp of de bestaande kennis bij soortgelijke toepassingen. In het transaktieschema staan termen als: het opzoeken van naam, adres, woonplaats en bedrag, het ophalen van alle gewenste artikelregels, het opzoeken van orderregels met een levertermijn langer dan een maand, het verzamelen van gegevens voor het faktuuroverzicht. Het transaktieschema is en blijft een gebruikersdokument. Wanneer dit soort omschrijvingen voor hem voldoende zijn, zal ergens anders een keer de detaillering en de kwantificering plaats moeten vinden. Dit behoort tot het werk van de transaktie-analist, omdat hij die gegevens verwerkt in het detailschema.
De resultaten van deze vorm van Transaktie analyse zijn de verwachte technische resultaten. Wanneer we even kijken naar responsetijden, dan ontstaan er hier de eerste indikaties over. Transaktie analyse geeft de gemiddelde responsetijden weer. Wanneer in een transaktie slechts enkele interakties of maar een paar kritische interakties voorkomen, kan de informatie-analist altijd aan de transaktie-analist vragen om naast de gemiddelde responsetijd, even een paar interakties te specificeren. Deze vorm van Transaktie analyse levert de verwachte waarde op van de volgende technische attributen:

- I/O`s per transaktie
- I/O`s per seconde
- interakties per uur
- gemiddelde responsetijden
- printregels per uur.

Afgezien van de cijfers van responsetijden, is geen van deze cijfers van direkt belang voor de gebruikers. Het is van belang dat de informatie-analist zich steeds realiseert dat alle resultaten van Transaktie analyse uiteindelijk gebaseerd zijn op zijn transaktieschema.
Als ook de hoeveelheid verkeer in kaart moet worden gebracht, zou er een logische terminal moeten worden vastgesteld. In de praktijk betekent dat uitgaan van de beschikbare beeldschermen in een P3-omgeving of uitgaan van een 3270-achtig beeldscherm met zonodig als alternatief een dom start-stop beeldscherm in een P4-omgeving.
- Technische Transaktie analyse. Deze analyse wordt uitgevoerd tijdens het technische ontwerp. Dan zijn de details van de verwerking bekend, en ook de hardware en de besturing van beeldschermen. Dan is ook bekend of er een netwerk moet worden ontworpen en welke gegevens daarvoor nodig zijn. Deze vorm van analyse levert de definitieve cijfers voor de technische attributen en de netwerkattributen. Een puur technisch gebeuren dus, waarvan voor de gebruiker alleen de uiteindelijke responsetijden en eventuele wijzigingen in de dialoog tengevolge van de hardware van de beeldschermen van belang zijn. Deze zaken moeten worden doorgegeven aan de informatie-analist. Anders gezegd: de informatie-analist moet aan het eind van het technisch ontwerp bij de transaktie-analist kontroleren of wat de gebruiker betreft, alles nog in overeenstemming is met het transaktieschema en de dialoogsimulatie.
De drie vormen van Transaktie analyse kunnen elkaar opvolgen. Vroeg in het logisch ontwerp wordt de ergonomische analyse uitgevoerd om de sociale aspekten in een zo vroeg mogelijk stadium te kunnen bespreken met de gebruikers. Later in het logisch ontwerp worden de beschikbare detailschema`s uitgebreid met de verwerkingsaspekten en opnieuw door het rekenprogramma gehaald. Tenslotte worden tijdens het technisch ontwerp de technische gegevens van beeldschermen en verwerking opgenomen in de bestaande detailschema`s.
Wanneer er tijdens het logisch ontwerp niets gedaan is aan transaktie-ontwerp en men wil tijdens het technisch ontwerp of tijdens de bouw alsnog Transaktie analyse uitvoeren dan kunnen op basis van transaktieschema`s ook direkt de technische detailschema`s gemaakt worden en de technische analyse worden uitgevoerd.
Naast deze indeling in vormen van Transaktie analyse bestaan er

Transaktieschema centraal

Transaktienaam: Wijziging prijslijst.

Menselijke handelingen	Transport	Machinale verwerking
Het opvragen van alle artikelen van een leverancier		
	-----)	Displayen van schermen met alle artikelen van een leverancier.
	(-----	
Het intoetsen van alle prijswijzigingen	-----)	Het verwerken van de wijzigingen.

Fig. 44.2 Een voorbeeld van een globaal transaktieschema.

nog een paar graden van nauwkeurigheid. Wanneer men tijdens het vooronderzoek toch een indruk wil hebben van het aantal terminal-uren op diverse werkplekken, kan er een globale ergonomische Transaktie analyse worden uitgevoerd. Dat betekent dat het transaktieschema globaal de menselijke handelingen beschrijft en het principe van de verwerking door de computer. Zo zou Fig. 44.2 de globale versie van Fig. 44.3 kunnen zijn. De transaktie-analist vraagt dan om een schatting van het aantal wijzigingen per leverancier, het aantal posities van een prijs en het aantal artikelregels per leverancier. De resultaten van deze globale Transaktie analyse zijn uiteraard globaal, maar in het vooronderzoek zijn globale cijfers vaak al heel verrassend. De konklusies van Transaktie analyse, die in paragraaf 44.6 getrokken worden, kunnen dan in grote lijnen en weliswaar minder nauwkeurig, al in het vooronderzoek aanleiding geven bepaalde mogelijkheden als onhaalbaar te bestempelen. Wanneer tijdens het logisch ontwerp wel transaktieschema's zijn gemaakt, maar nog geen dialoogsimulatie is uitgevoerd, kan er op basis van die voorlopige transaktieschema's Transaktie analyse worden uitgevoerd. Wanneer dialoogsimulatie is uitgevoerd, ligt de dialoog en de bijhorende schermlay-out vast en kan het definitieve transaktieschema worden gemaakt, onafhankelijk van de fase waarin het projekt verkeert. Als dialoogsimulatie tijdens het vooronderzoek wordt gedaan ontstaat tijdens

het vooronderzoek het definitieve transaktieschema, nauwkeurig genoeg voor een ergonomische Transaktie analyse. Op dialoogsimulatie kan dus nooit een globale ergonomische Transaktie analyse volgen. Samenvattend: de vormen van Transaktie analyse hebben te maken met de gewenste of haalbare resultaten. De nauwkeurigheid van de resultaten hangt af van de nauwkeurigheid waarmee kwantiteiten kunnen worden weergegeven op een detailschema: garbage in, garbage out.
Dan nu nog iets over de interface tussen informatie-analisten en transaktie-analisten via transaktieschema's.

44.3 Kwantiteiten binnen transakties

Zoals in een vorige paragraaf werd opgemerkt is het transaktieschema in de eerste plaats een gebruikersdokument en in de tweede plaats het startdokument voor transaktie-analist. Daarom kan het geen kwaad de inhoud van een transaktieschema ook te richten op de transaktie-analist, als er maar geen afbreuk wordt gedaan aan de primaire funktie. Ook al maken gebruikers zelf transaktieschema's, dan is er nog niets tegen om het daarna in goed overleg, zo bruikbaar mogelijk voor de transaktie-analist te maken. We nemen een eenvoudig transaktieschema als voorbeeld, zie Fig. 44.3. Er moeten prijswijzigingen worden ingevoerd. Het systeem is niet georganiseerd op artikel van de leverancier, maar op basis van een eigen artikelnummer. Per leverancier komen regelmatig nieuwe prijsoverzichten binnen. Soms moet ook de omschrijving van de artikelen worden aangepast. Laten we aannemen dat dit transaktieschema voor de gebruiker duidelijk genoeg is en geaksepteerd als beschrijving van de transaktie. Overigens is het schema er maar een uit een serie en is de gebruiker gewend geraakt aan termen als masker, displayen en menu, als hij die al niet kende van de voorbereidende opleiding (15). Kennelijk is er een menuscherm ontworpen, waarvan PRIJSWIJZIGING een van de mogelijkheden is. We zullen nu het transaktieschema eens lezen door de bril van een transaktie-analist, die er een detailschema van moet maken.
- Het kiezen van een funktie. Als aanloop voor de transaktie moet gekozen worden uit een serie mogelijkheden. Dat betekent voor de computer het selekteren van het menuscherm, het verwerken van de keuze. De gebruiker typt alleen de keuze in en wacht op het verschijnen van het eerste scherm. De transaktie-analist vraagt zich bij alles af, of het kwantitatief belangrijk is. Als een beeldscherm bijna alleen voor een transaktie wordt gebruikt verschijnt het menuscherm dus praktisch nooit en wordt er ook geen keuze ingetypt. Als het intypen van de keuze wel interessant is, wil de transaktie-analist nog weten hoeveel toetsen er moeten worden aangeslagen. Voor de transaktie-analist had die zin dus ook mogen

Transaktieschema centraal

Transaktienaam: Wijzigen prijslijst.

Menselijke handelingen	Transport	Machinale verwerking
	(-----	Het selekteren van het menu en displayen
Het kiezen van de funktie: prijswijziging	-----)	Het selekteren van het masker en displayen
	(-----	
Het lezen en intypen van de leverancierskode	-----)	Het zoeken van alle artikelregels, sorteren en displayen van de eerste serie artikelregels.
	(-----	
Bladeren tot het juiste artikel is gevonden. Het kontroleren, eventueel wijzigen van de omschrijving, het wijzigen van de prijs	-----)	Het verwerken van de wijzigingen, het eventueel displayen van de volgende serie regels.

Fig. 44.3 Een eenvoudige transaktie.

luiden: Het intoetsen van `PW` voor het kiezen van de funktie: PRIJSWIJZIGING, voor 10 tot 20 prijswijzigingen. De transaktie-analist geeft nu op in zijn detailschema, het selekteren van een menuscherm, het aantal in te toetsen tekens en de kans daarop. Het rekenprogramma rekent dat om naar een gewogen hoeveelheid verwerking en aantal in te toetsen tekens per transaktie.
- Het selekteren van het masker. Bij een ergonomische Transaktie analyse is het aantal tekens van een schermlay-out niet van belang. Wanneer het later, tijdens het technisch ontwerp, gaat om netwerkberekeningen wordt het detailschema op dit soort punten verder gedetailleerd.
- Het lezen en intypen van de leverancierskode. Ook hier zal de transaktie-analist zich weer afvragen om hoeveel tekens het gaat. Daarom zou er best een afspraak gemaakt kunnen worden over het aangeven van lengtes van velden. De transaktie-analist zet bijvoorbeeld achter ieder veld tussen haakjes de lengte en desgewenst het soort veld. De zin kan er dan als volgt uitzien: Het lezen en intypen van de leverancierskode(8).
- Het zoeken van alle artikelregels. Bij een ergonomische Transaktie analyse zal de transaktie-analist de verwerking zodanig kwantificeren dat er een of twee seconden uit de responsetijd komt. Als duidelijk is dat het om een langdurig verwerkingsproces gaat, kan hij er voor zorgen dat er 10 seconden responsetijd ontstaat. Hoe het ook zij, de verwerking wordt niet gedetailleerd, er worden responsetijden ingesteld.
Als het gaat om een logische Transaktie analyse zal de transaktie-analist zich afvragen hoeveel artikelregels er bij een leverancier horen en hoeveel er op een scherm mogen verschijnen. Wat hem betreft had de zin op het transaktieschema bijvoorbeeld mogen luiden: Het zoeken van alle 100 +/- 20 artikelregels, sorteren en displayen van de eerste 15 regels. De specifikatie van het aantal artikelregels had natuurlijk ook mogen luiden: Gem. 100, min. 50, max. 200. Daarna kan zich, wanneer het om een cruciaal aspekt blijkt te gaan, nog een diskussie ontwikkelen rond de verdeling. Transaktie analyse maakt het mogelijk om met gemiddelde en varianties te werken, maar bij niet normale verdelingen kan heel snel de worst case naast de gemiddelde situatie bekeken worden, omdat na een enkele wijziging het rekenprogramma de hele transaktie opnieuw doorrekent.
- Bladeren tot het juiste artikel is gevonden. Gegeven de gesorteerde regels, is het mogelijk te bepalen hoeveel er gemiddeld gebladerd zal worden, zonodig proefondervindelijk. Er had bijvoorbeeld mogen staan: Gem. 3 keer bladeren tot het juiste artikel is gevonden.
- Het wijzigen van de omschrijving. De transaktie-analist wil hier natuurlijk weten in hoeveel procent van de gevallen dat ge-

beurt en hoeveel tekens er dan worden ingetoets. De tekst had bijvoorbeeld kunnen luiden: Het kontroleren van de artikelomschrijvingen en in circa 10% van de gevallen het intypen van een nieuwe omschrijving (30 +/- 10 tekens).
- Het verwerken van de wijzigingen en het eventueel displayen van de volgende serie regels. Bij een ergonomische Transaktie analyse zal de transaktie-analist een passende responsetjd bedenken, bij een logische Transaktie analyse kan hij rekenen met een update van ieder artikel waarvan de prijs is gewijzigd.
Met dit voorbeeld is voldoende aangegeven dat de informatie-analist er verstandig aan doet zich te realiseren, dat een transaktieschema een keer vertaald moet worden naar een detailschema. Hoewel hij dat schema zelf niet maakt zal hij of de gebruiker toch de kwantiteiten moeten leveren. Hij kan dus zoveel mogelijk cijfers verwerken in het transaktieschema en als hij dat niet doet komt de transaktie-analist er toch een keer om vragen. Voor alle kwantiteiten geldt, dat hij steeds in het oog moet houden of ze echt belangrijk zijn. In twijfelgevallen kunnen even twee waarden worden geprobeerd om te kijken hoe groot de invloed van een bepaald aspekt is. Blijkt de invloed groot te zijn, dan is het de moeite waard om uit te zoeken wat een praktisch getal is. Een voorbeeld zijn de denk- en wachttijden. In een transaktie moet een gebruiker bij een veel voorkomende transaktie de op het scherm getoonde regels kontroleren. Deze denktijd zal vaak geschat worden. In een situatie waarin een dergelijke transaktie een paar honderd keer per dag op een tiental beeldschermen moet worden uitgevoerd, kan dit cijfer van belang zijn voor de berekening van het aantal beeldschermen en de systeembelasting. In zo'n geval is een meting van die tijd gedurende een dialoogsimulatiesessie, letterlijk de moeite waard.
In gevallen waarin twijfels bestaan omtrent de verdeling van niet konstante grootheden kan gemakkelijk een worst case worden berekend naast de gemiddelde situatie. Soms is het verstandig een transaktie in twee transakties te splitsen. Laten we als voorbeeld nemen een orderentry-situatie waarin sprake is van een gemiddeld aantal orderregels en een bepaalde spreiding, maar waarin ook bulkorders voorkomen met een veelvoud van het gemiddelde aantal orderregels. De gebruiker zal het verschil tussen de gewone orders en de bulkorders erg goed kennen. De gemiddelde automatiseerder heeft geen belangstelling voor het verschil omdat beide transakties door hetzelfde programma worden verwerkt. Een echte transaktie-ontwerper brengt de verschillen in kaart, omdat dan de konsekwenties zowel voor de gebruiker als voor het systeem bespreekbaar worden. Hij maakt twee transaktieschema's, een met de naam NORMALE ORDERS en een met de naam BULKORDERS. Alleen de aanduiding van het aantal orderregels per order verschilt, voor de

rest zijn de transaktieschema's gelijk.
Ter inleiding op de volgende paragraaf zullen we bespreken hoe de kwantiteiten in het detailschema moeten worden opgenomen. Zaken, waarvan de informatie-analist kan verwachten dat de transaktie-analist naar cijfers zal vragen. We hebben in het voorbeeld van Fig. 44.3 al het een en ander gezien. Het is duidelijk dat er in iedere transaktie iets ingetypt wordt. Er zal ook altijd iets door de computer gedaan worden. Bij Transaktie analyse wordt iedere transaktie afgebroken in een aantal standaardelementen. Het intypen is zo'n element. Deze gestandaardiseerde elementen noemen we parameters. Iedere parameter wordt op het detailschema aangegeven met een kode, zodat het rekenprogramma ermee kan werken. Wanneer op het transaktieschema staat: Het intypen van de leverancierskode (8), dan zal er op het detailschema staan: Intypen debiteurennummer, 1, 9, 0. De tekst op het detailschema is meestal een telegramstijlversie van de tekst op het transaktieschema. Het cijfer 1 is de parameterkode voor intypen. Het cijfer 9 geeft het aantal toetsen aan: 8 plus een afsluittoets. De nul geeft aan dat er geen spreiding is. Het lezen van de leverancierskode valt in dit geval onder denk- en wachttijden. Bij konstant intypen van van reeksen velden wordt meestal blind getypt en zal er geen sprake zijn van een denk- en wachttijd per te lezen veld. Dit soort aspekten wordt behandeld in (17). Denk- en wachttijden hebben parameterskode 12.
Wanneer in het transaktieschema staat: Het lezen en intypen van de leverancierskode (8), verschijnen er in het detailschema twee regels:
- Het lezen van de leverancierskode, 12, 1, 2, 1.
- Het intypen van de leverancierskode, 1, 1, 9, 0.

Na de tekst staan in het detailschema vier cijfers: de parameterkode, de kansfaktor, de gemiddelde waarde van de parameter en zijn variantie. In het vorige voorbeeld was de kansfaktor nog even weggelaten. Er worden in het detailschema geen eenheden aangegeven, maar daaraan heeft niemand meer behoefte als hij een keer een detailschema heeft gemaakt of gelezen. Parameterkode 12 wordt uitgedrukt in seconden. Vuistgetallen voor dit soort parameters staan in (2), het handboek voor de transaktie-analisten.

44.4 Ergonomische parameters

We zullen nu de parameters behandelen die van belang zijn voor een ergonomische Transaktie analyse. In die analyse gaat het hoofdzakelijk om kwantificering van gebruikershandelingen. De verwerking door de computer wordt niet meegenomen, er wordt bijvoorbeeld een gemiddelde reponsetijd ingesteld voor alle transakties. Als er werkelijk grote variaties in responsetijden worden

verwacht, kan de transaktie-analist die uiteraard ook laten verschijnen.
In de vorige paragraaf is aangegeven hoe de parameters worden opgenomen in een detailschema. We zullen nu het doel van de ergonomische parameters bespreken. Daarbij zal een enkele keer het woord terminaltransaktietijd (T.T.T.) vallen. Dat is een resultaat van Transaktie analyse en het geeft aan hoe lang een transaktie duurt. Het begrip wordt in de paragraaf Resultaten en Konklusies uitgebreid behandeld.
- Parameter 1: lengte van een in te toetsen veld, inclusief de afsluittoets. Deze parameter komt meestal vele malen voor in een transaktie en is bijna altijd belangrijk.
- Parameter 4: het aantal te displayen tekens. Meestal heeft de transaktie-analist dit gegeven niet nodig. Als hij het nodig heeft is dat tijdens het technisch ontwerp en dan is de schermlay-out bij hem ook bekend en kan hij het aantal tekens zelf bepalen.
- Parameter 5 en 7: een interaktie. In feite wordt met deze parameter de pijl naar rechts op het transaktieschema weergegeven. Er is maar een soort pijl naar rechts, maar er zijn twee kodes voor. In Transaktie analyse wordt onderscheid gemaakt tussen een dialoogresponse, parameterkode 5 en een afsluitresponse, parameterkode 7. Met een dialoogresponse wordt bedoeld een kritische interaktie. Het is voor de gebruiker storend wanneer die ineens erg lang duurt. Met afsluitreponse wordt een interaktie bedoeld die een hoeveelheid handelingen van de gebruiker afsluit. Geestelijk en/of lichamelijk leunt hij even achterover. Een lange responsetijd is dan misschien ook vervelend, maar het haalt hem niet zo uit zijn ritme of koncentratie, hij ontspant zich juist even. De transaktie-analist moet uit het verloop van de transaktie kunnen zien of een interaktie een afsluit- of een dialoogresponse is. De responsetijdeisen zoals die uit dialoogsimulatie komen moeten aansluiten bij dit onderscheid. Wanneer een informatie-analist ziet dat het om een zware verwerking gaat van een interaktie die eigenlijk voor de gebruiker een afsluitresponse is, moet hij niet een flitsende responsetijd eisen, maar op de simulator een lange responsetijd instellen. Juist met een middel als de dialoogsimulator heeft hij de mogelijkheid een, ook voor de gebruiker, redelijke eis te stellen, zie paragraaf 43.4.
Of in het detailschema kode 5 en kode 7 wordt gebruikt is dus ter beoordeling aan de transaktie-analist. Als hij twijfelt kan hij overleggen met de informatie-analist. Het rekenmodel telt alle kodes 5 en alle kodes 7 bij elkaar en houdt per kode de berekening van de responsetijden voor beide gescheiden. In het resultaat verschijnt een gemiddelde dialoogresponsetijd en een gemiddelde afsluitresponsetijd. Bij een ergonomische Transaktie analy-

se wordt de verwerking door de computer niet geanalyseerd of in kaart gebracht. De informatie-analist wil de resultaten van Transaktie analyse hebben bijvoorbeeld gebaseerd op een gemiddelde responsetijd van twee seconden. Dus de vraag zou kunnen zijn: hoeveel uren zit de gebruiker achter het beeldscherm bij 100 van deze transakties, uitgaande van een responsetijd van twee seconden? In dat geval kan de transaktie-analist in het model een bepaalde parameter zodanig instellen dat iedere responsetijd 2 seconden duurt. Dat betekent dat we de transaktie, zoals die ontworpen is met dialoogsimulatie, al kunnen evalueren met de gebruiker zonder dat de verwerking precies in kaart is gebracht. Daarbij blijft natuurlijk de vraag in hoeverre die twee seconden realistisch zijn. Maar de responsetijden vormen meestal slechts een fraktie van de totale terminal transaktietijd. In die situaties wordt er bij berekeningen die gebaseerd zijn op de terminal-transaktietijd geen grote fout gemaakt. Bovendien kost het bijna geen moeite om het rekenproces nog even de zaak door te laten rekenen bij responsetijden van bijvoorbeeld 5 seconden. Door de terminaltransaktietijden met elkaar te vergelijken ziet men of de responsetijd een kritisch element is in de terminaltransaktie-tijd.

- Parameter 11: aan- en uitloop. Uiteraard gaat het om aan- en uitloop binnen de transaktie. Het gaat om handelingen die vooraf-gaan aan het werken met het beeldscherm en die erop volgen. Een eenvoudig voorbeeld: de transaktie voor het opnemen van geld bij een bank. De transaktie begint met praten met de klant, een cheque overhandigen en het kontroleren van de cheque. Pas dan gaat de bankbediende met het beeldscherm werken. Dat is de aanloop binnen de transaktie. Wanneer de dialoog met de computer is afgelopen volgt er meestal een uitbetaling, soms voorafgegaan door een diskussie over het soort bankpapier. Dat is de uitloop binnen de transaktie. De volgende transaktie begint wanneer de volgende klant begroet wordt.

De aan- en uitloop is vaak een belangrijk onderdeel van de transaktie. Immers, hoe langer de aan- en uitloop hoe minder de transaktie het netwerk en het systeem belast. Dialoogsimulatie biedt de mogelijkheid om tijdens het werken met de simulator dit soort tijden te meten. De informatie-analist doet er goed aan die gemeten tijden te vermelden in het transaktieschema. Doet hij dat niet dan zal de transaktie-analist er naar komen vragen of zelf schattingen doen, soms gebaseerd op vuistgestallen uit (2). Parameter 11 wordt uitgedrukt in seconden.

- Parameter 12: denk- en wachttijden. Hier gaat het om menselijke handelingen binnen de dialoog, zoals het kontroleren van gegevens op het scherm, het vergelijken van gegevens op het scherm met gegevens op brondokumenten, het omslaan van een blad. Ook parameter

12 wordt uitgedrukt in seconden.
Het maakt voor de resultaten overigens niets uit of een bepaalde handeling nu in parameter 11 of 12 wordt uitgedrukt. Binnen een transaktie worden alle kodes 11 bij elkaar geteld en alle kodes 12. Beide totalen maken deel uit van de terminaltransaktietijd en het maakt voor dat resultaat niet uit of een handeling nu onder 11 of onder 12 is uitgevoerd in het detailschema. Het blijft natuurlijk verstandig om parameters te gebruiken zoals ze bedoeld zijn.
- Parameters 13: de intiksnelheid. In veel transakties is het intoetsen een belangrijk element en dus is deze parameter 13 belangrijk. De waarde zal uiteindelijk van de gebruiker moeten komen. Een typediploma betekent een snelheid van twee toetsen per seconde. Er zijn datatypistes die zes aanslagen per seconde halen. Dat zijn echter uitzonderingen. Vier aanslagen per seconde is al hoog. Beginnende gebruikers die in de handmatige situatie formulieren invullen, zullen een halve aanslag per seconde misschien nog niet halen. Hoeveel er ook ingetoetst wordt tijdens een transaktie, de intiksnelheid is slechts een parameter. Met andere woorden, het is heel eenvoudig het rekenprogramma even de resultaten te laten doorrekenen voor een halve, twee en vier aanslagen per seconde. In de eerste plaats is dat een gevoeligheidsanalyse: hoeveel invloed heeft de intiksnelheid op de uren achter het beeldscherm of op het aantal beeldschermen of op de bezetting van aanwezige beeldschermen? Blijkt die invloed aanzienlijk te zijn dan kan in de tweede plaats bijvoorbeeld de gebruikers worden voorgerekend met hoeveel uitloop van werk naar de volgende dag hij in het begin rekening moet houden. Want de gebruiker die beginnen met een halve aanslag per seconde zullen, zeker bij numerieke gegevens toch vrij snel hun intiksnelheid kunnen verhogen.
- Parameter 14: het foutherstelpercentage. Dit is het percentage van de invoertijd, nodig voor het herstellen van fouten. Dit percentage hangt af van de rubrieklengte en het percentage aanslagfouten. In (2) is een matrix opgenomen voor de bepaling van dit percentage. In de praktijk is 5% een redelijke aanname om mee te beginnen.
- Parameter 16: effektiviteitsfaktor. Wanneer een transaktie een minuut duurt, zal niemand verwachten dat er per dag, bij een achturige werkdag, 8 x 60 = 480 transakties per dag worden uitgevoerd. Er bestaat immers zoiets als pauzes, persoonlijke verzorging, praatje met een collega, koffie halen en drinken. Wanneer er per dag 480 transakties van een minuut worden uitgevoerd, kan dat niet met een beeldscherm. Een gebruikelijke effektiviteitsfaktor is 70%. Het aantal beeldschermen moet dus vermenigvuldigd worden met een faktor 1/0,7. In het voorbeeld wordt het dus circa

anderhalve terminal, in de praktijk meestal afgerond naar 2 stuks. Maar, omdat koffie drinken ook heel goed is voor de belasting van het systeem, berekent het rekenprogramma naast een nettoterminaltransaktietijd, ook een brutoterminaltransaktietijd. Het verschil tussen beide is dus de effektiviteitsfaktor. De brutoterminaltransaktietijd kan gebruikt worden om het aantal beeldschermen en de gemiddelde systeembelasting te berekenen, de nettoterminaltransaktietijd om het aantal transakties per uur in een pieksituatie te berekenen. Wanneer een piek niet langer duurt dan ongeveer een uur, is immers de effektiviteit 100%. Zo kan een transaktie-analist de gemiddelde systeembelasting berekenen en de belasting gedurende een piek.
De effektiviteitsfaktor is moeilijk precies te bepalen. Dat is ook niet nodig: zeventig procent is een algemeen geldende waarde. Wanneer gebruikers willen bezuinigen op het aantal beeldschermen bijvoorbeeld door te stellen dat de effektiviteitsfaktor op een bepaalde afdeling hoger is dan het algemene gemiddelde, dan kan dat. De faktor wordt op bijvoorbeeld 80% gesteld. Die waarde ligt nu dus vast en wordt bij de resultaten vermeld. Wanneer nu achteraf blijkt dat met de beschikbare terminals toch het werk dat per dag gedaan moet worden, niet gerealiseerd kan worden, dan ligt van de mogelijke oorzaken er in ieder geval een duidelijk vast. De boven het gemiddelde uitstekende werklust van de gebruikers is vastgelegd en heeft tot konsekwenties geleid voor het aantal beeldschermen. Het blijft natuurlijk noodzakelijk om de uiteindelijke situatie zo goed mogelijk weer te geven. Wanneer er gekozen zal worden voor een oplossing waarin de typistes elkaar in groepjes aflossen bij de terminals dan is er sprake van een effektiviteitsfaktor van 100%!
Tenslotte nog een algemene opmerking over deze faktor. Het gaat om een toeslag op de nettotransaktietijd tengevolge van het niet produktief zijn van de gebruiker. Per definitie slaat het percentage dan ook alleen op de menselijke handelingen. Wanneer een transaktietijd voor de helft uit reponsetijd zou bestaan, is een verhoging van de effektiviteit van de gebruiker maar voor een deel van invloed op de transaktietijd. Koffiedrinken, dat tijdens de responsetijd plaatsvindt nalaten, verbetert de effektiviteitsfaktor niet. Enige voorzichtigheid in het gebruik van de faktor is dus geboden, maar in de praktijk zullen transaktietijden zeker voor meer dan de helft bepaald worden door menselijk handelen. De resultaten van Transaktie analyse geven drie verhoudingen direkt weer. Op basis daarvan kan men dus zo nauwkeurig mogelijk verder gaan met de konklusies of iteraties, als de nauwkeurigheid van de gegevens over gebruikerssituatie maar mogelijk maken. Het aantal parameters is groter dan die tot nu toe werden besproken. De overige parameters worden gebruikt door de transaktie-

analist. De behandelde parameters zijn van belang voor de informatie-analist omdat de kwantiteiten die erbij horen bij hem of bij de gebruiker vandaan moeten komen. Er is natuurlijk ook niets tegen dat een informatie-analist zelf het detailschema maakt voor een ergonomische Transaktie analyse. Hij moet alleen de transaktie-analist laten zorgen voor de gewenste responsetijden.
Wanneer het detailschema gereed is, is er eigenlijk een rekenmodel van de transaktie ontstaan. Van dat model worden door het rekenprogramma de resultaten bepaald. Door het wijzigen van een parameter kan heel snel vastgesteld worden of een element in de transaktie veel invloed heeft op het resultaat. Als dat zo is, loont het de moeite de waarde van dat element zo nauwkeurig mogelijk te bepalen.

44.5 Kwantiteiten van transakties

In de voorgaande paragrafen ging het over kwantiteiten binnen transakties. Wanneer het detailschema is ingevoerd in het rekenprogramma verschijnen er drie pagina's output per transaktie. Daarmee is voor een transaktie Transaktie analyse afgelopen: de cijfers zijn beschikbaar. Voor de informatie-analist begint dan het eigenlijke werk: het trekken van konklusies. Daarbij zijn cijfers nodig over die transakties. Als Transaktie analyse een terminaltransaktietijd oplevert van 120 seconden maar er is niet bekend om hoeveel transakties per dag het gaat, valt er weinig te konkluderen.
De informatie-analist moet weten welke resultaten Transaktie analyse oplevert en waarom hij de analyse uitvoerde. Dat laatste zal hem ook duidelijk maken naar welke kwantiteiten hij moet vragen. Er zijn geen regels voor te geven. Onderstaande opsomming moet dan ook meer gezien worden als een aantal voorbeelden.
- Het aantal transakties per dag.
- De pieken. Daarbij kan het gaan om piekuren, piekdagen binnen een week of aan het eind van een periode of piekmaanden ten gevolge van seizoeninvloeden.
- De doorlooptijd. Bestaat er een tijdstip waarop bepaalde transakties uitgevoerd moeten zijn? Welke overloop naar een volgende dag is toegestaan?
- De verdeling van transakties over de werkplekken of groepen van werkplekken. Is overloop van de ene werkplek of -groep naar een andere mogelijk?
- Het aantal medewerkers dat betrokken is bij een transaktie. Dit getal kan gebruikt worden ter kontrole van de eigen konklusies uit Transaktie analyse. Wanneer er volgens die konklusies meer beeldschermen nodig zijn dan er nu medewerkers zijn, is er misschien wel iets aan de hand, hoewel er situaties zijn waarin

die verhouding best mogelijk is. Tenslotte nog iets over het moment waarop de kwantiteiten verzameld kunnen worden. Er vindt binnen een projekt natuurlijk al kommunikatie plaats met de gebruikers. In de allereerste analysegesprekken komen soms spontaan al cijfers boven water. Het kan geen kwaad die alvast te noteren. In verband daarmee zou het goed zijn, vanaf het begin van een projekt een struktuur te bedenken waarin kwantiteiten van transakties en kwantiteiten binnen transakties worden opgeborgen. Dan zijn ze voor de volgende fasen beschikbaar voor Transaktie analyse, voor transaktie-analisten en systeemontwerpers.
Tijdens de analysefase worden dus de cijfers genoteerd die spontaan gegeven worden, tijdens de ontwerpfase moet precies bekend zijn welke gegevens er nog meer nodig zijn. Het is van belang de gebruiker bijvoorbeeld via een presentatie over de konklusies uit Transaktie analyse duidelijk te maken waarom de cijfers nodig zijn en wanneer er naar gevraagd zal worden. Sommige cijfers vragen namelijk van de gebruiker ook enige voorbereiding. Niets is zo vervelend voor een gebruiker als het op allerlei verschillende momenten worden aangeschoten voor cijfers, door verschillende automatiseerders.

44.6 Ergonomische resultaten en konklusies

Per detailschema levert het rekenprogramma drie pagina's resultaten.
- Het parameteroverzicht. Per parameter het totaal aantal eenheden in gemiddelde waarde en spreiding.
- De terminaltransaktietijd. Dit is een berekening van de transaktietijd, met de samenstellende elementen.
- De lijn- en responsetijdaspekten. Deze pagina bevat gegevens over verkeer en responsetijden.

Terwille van de leesbaarheid zijn de output-pagina's van het rekenprogramma niet afgedrukt van het printpapier, maar opnieuw getypt.
Het programma voegt niets toe aan de cijfers van het detailschema. Alle berekeningen worden uitgevoerd op basis van de totalen zoals die voorkomen op het parameteroverzicht. Dat parameteroverzicht is de gewogen optelling per parameter, van de cijfers van het detailschema. De weging vindt plaats via de kansfaktoren in het detailschema.
Enkele parameters zoals de typesnelheid, zijn meestal niet ontstaan door cijfers op te tellen: de parameter voor de typesnelheid komt maar een keer voor op het detailschema. Fig. 44.4 is een voorbeeld van een parameteroverzicht. De parameters 1, 4, 5, 7, 11, 12, 13 en 15 zijn behandeld in de paragraaf Ergonomische parameters. De overige worden behandeld in het deel voor de

Parameter-overzicht

Code	Rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:				
1.	Invoerlengte	76.03	Aanslagen	278.03
2.	Transportlengte invoer	97.27	Tekens	362.78
3.	Transportlengte uitvoer	70.74	Tekens	36.55
4.	Uitvoerlengte	208.74	Tekens	558.59
5.	Dialoogresponses	6.71	Responses	1.21
6.	Dialoogresponse eenheden	14.01	Eenheden	4.09
7.	Afsluitresponses	1.00	Responses	0.00
8.	Afsluitresponse eenheden	10.00	Eenheden	4.00
Personele-parameters:				
11.	Aanloop en uitloop	2.50	Seconden	0.60
12.	Wachttijden en denktijden	3.00	Seconden	3.00
13.	Intiksnelheid	2.50	Tek./sec.	1.00
14.	Foutherstelpercentage	5.00	Procent	2.40
15.	Min. invoer repetitietijd	0.00	Seconden	0.00
16.	Effektieve werktijd	70.00	Procent	100.00
Machine-parameters:				
21.	Transportvertraging	1.20	Seconden	1.00
22.	Tijd per response eenheid	0.10	Seconden	0.01
23.	Afdruksnelheid	25000.00	Tek./sec.	0.00

Fig. 44.4 Voorbeeld van het parameteroverzicht

transaktie-analist. Dat laatste geldt trouwens ook voor de pagina lijn- en responsetijdaspekten.
De ergonomische parameters op het parameteroverzicht moeten altijd even worden doorgenomen. Op basis van deze cijfers worden de berekeningen voor de andere twee pagina's uitgevoerd en een typefout in het detailschema kan vervelende gevolgen hebben. Het is natuurlijk onmogelijk grenzen aan te geven waarbinnen de waarden moeten liggen. Een informatie-analist kent echter het transaktieschema en als het goed is, heeft hij daarop al wat kwantiteiten aangegeven. Alles op het parameteroverzicht, wat vraagtekens oproept kan direkt op het transaktieschema worden gekontroleerd. Als dat het probleem niet oplost, kan het detailschema worden bekeken. Het detailschema zoals dat door het rekenprogramma wordt afgedrukt bevat voor alle regels waarvan de kansfaktor kleiner is dan 1, de gewogen waarde. Dat maakt een snelle kontrole mogelijk.
De pagina terminaltransaktietijd levert de ergonomische resultaten op van de analyse. Fig. 44.5 is een voorbeeld van zo'n pagina. De nettoterminaltransaktietijd is de som van alle, naar seconden omgerekende menselijke handelingen en de verwerking door het systeem. De nettoterminaltransaktietijd is de tijd die nodig is om een transaktie uit te voeren zoals die beschreven is op het transaktieschema. Die tijd wordt gedeeld door de effektiviteitsfaktor en dat levert de brutoterminaltransaktietijd op. Wanneer het aantal benodigde beeldschermen wordt berekend op basis van de terminaltransaktietijd, is de brutotijd de basis voor het aantal beeldschermen. In de paragraaf Voorbeelden van ergonomische konklusies (44.7) zal een aantal toepassingen van de terminaltransaktietijd gegeven worden. Of de netto- of de brutotijd gebruikt moet worden, hangt van de situatie af en dat moet de informatie-analist zelf kunnen bepalen na de uitleg in de vorige paragraaf.
We zullen nu de elementen van de transaktietijd kort behandelen. Een uitvoerige bespreking heeft plaats in (17). Zie Fig. 44.5. In de eerste plaats wordt de invoertijd berekend. Het totaal aantal aanslagen zoals dat te vinden is op het parameteroverzicht bij parameter 1, wordt gedeeld door de typesnelheid, parameter 13. Het volgende element in de terminaltransaktietijd is de denk- en wachttijden. Dit is parameter 12 uit het parameteroverzicht. Dan komen er twee elementen die te maken hebben met het transport van het verkeer door het netwerk en met de verwerking door de computer. Het eerste element is de TRANSPORTTIJD, het tweede de VERWERKINGSTIJD. Deze aspekten worden behandeld in het deel voor de transaktie-analist. Bij een ergonomische analyse kan de informatie-analist aan de transaktie-analist vragen er voor te zorgen dat in het model wordt uitgegaan van een bepaalde responsetijd. Wanneer de informatie-analist bijvoorbeeld zou vragen uit te gaan van twee seconden responsetijd voor iedere interaktie, dan kan de

Terminal Transaktie tijd (T.T.T.)

Rubriek	Tussen-waarden		Aantal seconden	Variantie
Invoerlengte	76.04			
Intiksnelheid	2.50			
	--------	/		
Invoertijd			30.41	192.46
Wachttijden en denktijden			3.00	3.00
Responses	7.71			
Transportvertraging	1.20			
	--------	*		
Transporttijd			9.20	61.22
Response-eenheden	24.01			
Tijd per eenheid	0.10			
	--------	*		
Verwerkingstijd			2.40	5.85
Uitvoerlengte	208.74			
Afdruksnelheid	25000.00			
	--------	/		
Uitvoertijd			0.01	0.00
			----------	----------
Subtotaal			45.08	262.53
Foutherstelpercentage	5.00			
Foutherstel-tijd			2.25	1.14
Aanloop en uitloop			2.50	0.60
			----------	----------
Netto T.T.T.			49.83	264.27
(Volledig effektief)	Standaardafwijking			16.26
Bruto T.T.T.			71.18	642.74
(Bij 70% effektief)	Standaardafwijking			25.35

Procentuele verdeling van de T.T.T.

Invoertijd	61.03 %
Wachttijden en denktijden	6.02 %
Dialoogresponse-tijden	18.98 %
Afsluitresponse-tijden	4.42 %
Uitvoer-tijd	0.02 %
Tijd voor fout herstellen	4.76 %
Tijd voor aanloop en uitloop	5.02 %

Fig. 44.5 De pagina terminaltransaktietijd.

transaktie-analist dat realiseren door bepaalde parameters zo in te stellen, dat het programma rekent met een verwerkingstijd die gelijk is aan het aantal interakties maal twee seconden. Die tijd verschijnt dan achter VERWERKINGSTIJD. Het volgende element is de UITVOERTIJD. Deze tijd is bij een ergonomische analyse in een situatie met beeldschermen niet van belang omdat de tijd die nodig is om 2000 tekens vanuit de buffer in de terminal op het scherm te zetten, te verwaarlozen is. Bij keyboardprinters of hardcopyprinters is dit element wel van belang. Dan moet de transaktieanalist goed worden verteld hoe de printer is verwerkt in het transaktieschema. De transaktie-analist kan dan besluiten een apart detailschema te maken voor het printen of het printen te verwerken in hetzelfde detailschema als het beeldschermdeel van de transaktie. Wanneer voor dat laatste wordt gekozen, is UITVOERLENGTE het aantal te printen tekens van parameterkode 4 uit het parameteroverzicht. Deze waarde wordt gedeeld door de printsnelheid, parameterkode 23, en zo ontstaat de UITVOERTIJD, als onderdeel van een beeldschermtransaktie. Details van dit aspekt worden behandeld in het deel voor de transaktie-analist.
Het volgende element is de fouthersteltijd. Gegeven het percentage, aangegeven met parameterkode 14, wordt de tijd berekend die nodig is voor het herstellen van typefouten.
Dan volgt AANLOOP EN UITLOOP. Dit is de waarde van parameterkode 11 uit het parameteroverzicht. Daarmee is de NETTO T.T.T. bepaald. De gemiddelde waarde en de standaardafwijking worden afgedrukt. Wanneer geen rekening wordt gehouden met pauzes en persoonlijke verzorging, is dit de tijd die nodig is om een transaktie uit te voeren. Vervolgens berekent het programma de BRUTO T.T.T. Dat is de tijd die een transaktie duurt inclusief pauzes en persoonlijke verzorging. Uitgaande van een achturige werkdag, kan op basis van dit cijfer worden bepaald hoeveel transakties er per dag beeldscherm uitgevoerd kunnen worden. De terminaltransaktietijd is opgebouwd uit elementen die op zich vaak niet normaal verdeeld zijn. Wiskundig is te bewijzen dat het geheel van een aantal niet-normale verdelingen al gauw de normale verdeling benadert.
Tenslotte wordt op de pagina afgedrukt de procentuele verdeling van de terminaltransaktietijd. Als er dus problemen ontstaan door te lange transaktietijden dan is hier in een oogopslag te zien welke elementen van de transaktie de moeite waard zijn om nog eens kritisch onderzocht te worden. Wanneer de aan- en uitloop bijvoorbeeld 30% van de transaktietijd bedraagt, is het nuttig de aansluiting van het werken met het beeldscherm op de handmatige procedures nog eens te bezien.
Daarmee zijn de resultaten van een ergonomische Transaktie analyse behandeld. We zullen nu puntsgewijs een aantal aspekten van

deze cijfers bespreken. In de volgende paragraaf wordt een aantal toepassingen besproken.
- Het werken met gemiddelden en standaardafwijkingen maakt het mogelijk een goed overzicht te houden over het geheel van een ingewikkelde situatie. Toch blijft gelden dat we altijd attent moeten blijven op mogelijke fouten of onpraktische konklusies. Wanneer een transaktie op twee manieren kan verlopen, waarbij de ene versie 20 seconden duurt en de andere 140 seconden, dan zal, als ze allebei even vaak voorkomen, het programma een gemiddelde transaktietijd berekenen van 80 seconden. Dat is dan een nietszeggend cijfer, omdat er nooit een transaktie zal voorkomen die 80 seconden duurt. In zo'n geval had de informatie-analist moeten besluiten er twee transakties van te maken. Dat schept ook in de richting van de gebruiker een stuk duidelijkheid.
- Het kost weinig moeite het programma een worst case te laten doorrekenen. Er hoeven alleen maar wat parameters te worden aangepast en het nieuwe resultaat komt beschikbaar.
- Het is even eenvoudig om vast te stellen of een schatting verantwoord is of niet. Een andere waarde in het detailschema en de nieuwe resultaten geven aan of de transaktietijd gevoelig is voor die schatting. Als dat het geval is, moet de schatting nauwkeurig bepaald worden.
- Het is verrassend te zien hoe snel en effektief cijfers werken in een diskussie. Wanneer de gebruikers voorgerekend kan worden wat de gevolgen zijn van fouten in schattingen van aantallen transakties, zijn ze al gauw geneigd om nog eens een onderzoek uit te voeren naar de juistheid van hun cijfers. En daarmee is dat probleem neergelegd waar het hoort, bij de gebruikers.
- Per transaktie zijn nu de cijfers bekend. Dat betekent dat er nu nog een overzicht gemaakt moet worden van alle resultaten. We zullen zo'n overzicht van de ergonomische resultaten behandelen bij de voorbeelden in de volgende paragraaf. Die overzichten vormen de sociale aspekten in cijfers en dienen besproken te worden met gebruikers, personeelszaken en de ondernemingsraad. Tijdens de dialoogsimulatie hebben de gebruikers ervaren wat het beeldscherm voor hun werk betekent, het resultatenoverzicht levert inzicht in de tijdsbesteding per werkdag. Daarmee is per werkplek een kompleet beeld ontstaan van de sociale aspekten van de automatisering en dat is iets anders dan een vaag algemeen verhaal over chips en werkloosheid of allerlei bewustwordingsprocessen.

44.7 Voorbeelden van ergonomische konklusies

In deze paragraaf wordt een aantal voorbeelden van toepassingen van de ergonomische resultaten van Transaktie analyse gegeven. Vaak kunnen overzichten veel beter gemaakt worden op een micro-

Transaktie	Werkplek-groep	Aantal per dag	Bruto T.T.T.	Uren per dag
T1	WPGA	15	803,27	3.35
T2	WPGB	25	438,73	3
T3	WPGA/B	1166	42,40	13.73
T4	WPGA	52	41,13	0.60
T5	WPGD	360	239,66	24
T6	WPGA	10	31,38	0.087
T7	WPGE	10	17,30	0.048
T8	WPGA	8	60	0.13
T9	WPGA	4	60	0.06
T10	WPGA	16	71,43	0.32
T11	WPGA	12	78,57	0.26
T12	WPGF	360	42.86	4.28
T13	WPGG	30	216,40	1.80
T13	WPGF	10	216,40	0.60
T14	WPGF	360	42,86	4.30
T15	WPGF	40	744,77	8.27
T16	WPGF	40	187,60	2.08
T17	WPGG	60	67,14	1.12
T17	WPGF	20	67,14	0.37
T18	WPGG	20	50	0.28
T18	WPGF	10	50	0.14
T19	WPGG	60	854,80	10.90
T20	WPGG	20	21,43	0.12
T20	WPGF	10	21,43	0.06
T21	WPGB	200	118,38	6.58
T22	WPGA/B	20	164,44	0.91
T23	WPGB	200	24,63	1.37
T24	WPGB	200	24,50	1.36
T25	WPGA/B	10	22,07	0.06
T26	WPGA/B	2	88,30	0.05
T27	WPGA	1000	20,46	5.68
T28	WPGA/B	20	109,17	0.60
T29	WPGA/B	300	189,20	15.76
T30	WPGA	15	794,75	3.31

Fig. 44.6 Voorbeeld van de berekening van het aantal beeldschermen.

computer met een spreadsheet-programma. De invloed van een parameter op het geheel kan dan snel bepaald worden.

Voorbeeld 1. Bepaling van het aantal beeldschermen en beeldschermuren.

In dit voorbeeld gaat het om een logische Transaktie analyse, waarvan nu alleen de ergonomische resultaten worden behandeld. Nadat was vastgesteld welke procedures vervangen zouden worden door transakties, kwam de vraag naar voren op hoeveel beeldschermen er op de verschillende afdelingen geplaatst zouden moeten worden. Voor de betrokken transakties werd Transaktie analyse uitgevoerd. De terminaltransaktietijd (T.T.T.) uitgedrukt in seconden is voor iedere transaktie opgenomen in Fig. 44.6. Met de gebruikers is vastgesteld waar iedere transaktie zal worden uitgevoerd. De laatste kolom in de figuur is het produkt van het aantal transakties per dag en de bruto T.T.T., gedeeld door 3600. Per afdeling is een werkplekgroep gedefinieerd zoals in Fig. 44.6 is aangegeven met WPGn, waarin n de afdelingsnaam is. Bij de bepaling van het aantal beeldschermen moest worden uitgegaan van de volgende punten:

- op afdeling B moeten de volgende transakties tussen 8.00 en 12.00 uur zijn afgehandeld: T2, T21, T23 en T24.
- 's middags kunnen op de beeldschermen van afdeling B zonodig transakties van afdeling A worden uitgevoerd.

We zullen nu met behulp van de cijfers uit Fig. 44.6 het aantal beeldschermen per werkplekgroep bepalen, uitgaande van 8 werkuren per dag.

- WPGA
  Transaktie: T1, T4, T6, T8, T9, T10, T11, T27 en T30
  Aantal beeldschermuren: 3.35 + .6 + .087 + .13 + .06 + .32 + .26 + 5.68 + 3.31 = 13.79
  Aantal beeldschermen: 2
  Overkapaciteit: ca. 2 beeldschermuren
- WPGB
  Transakties: T2, T21, T23 en T24
  Aantal beeldschermuren: 3 + 6.58 + 1.37 + 1.36 = 12.31
  Aantal beeldschermen: 3
  Te kort: ca 1/2 beeldschermuur
- WPGA/B
  Transakties: T3, T22, T25, T26, T28 en T29 van afdeling A kunnen ook op afdeling B worden uitgevoerd.
  Aantal beelschermuren: 13.73 + .91 + .06 + .05 + .6 + 15.76 = 31.11
  Beschikbaar bij WPGB: 3 x 8 - 12.31 = 11.69,
  dus nog nodig op WPGA voor deze transakties: 31.11-11.69=19.42
  Overkapaciteit op WPGA, als berekend: 2 beeldschermuren

Aantal beeldschermuren op WPGA: 17.42
Extra beeldschermen op WPGA: 2
Te kort op WPGA: 1.42 beeldschermuren

- WPGD
  Transaktie: T5
  Aantal beeldschermuren: 24
  Aantal beeldschermen: 3
- WPGE
  Transaktie: T7
  Aantal beeldschermuren: .048
  Aantal beeldschermen: 1
  Overkapaciteit: ca. 8 beeldschermuren(!)
- WPGF
  Transakties: T12, T13, T14, T15, T16, T17, T18 en T20
  Aantal beeldschermuren: 4.28 + .6 + 4.3 + 8.27 + 2.08 + .37 + .14 + .06 = 20.1
  Aantal beeldschermen: 3
  Overkapaciteit: ca. 2 beeldschermuren
- WPGG
  Transakties: T13, T17, T18, T19 en T20
  Aantal beeldschermuren: 1.8 + 1.2 + .28 + 10.9 + .12 = 14.22
  Aantal beeldschermen: 2
  Overkapaciteit: ca. 1.8 beeldschermuur.

Enige opmerkingen bij de resultaten.

- De bruto T.T.T. is in het voorbeeld de gemiddelde terminal-transaktietijd. Het aantal berekende beeldschermen is dan ook slechts het gemiddelde aantal. Het aantal beeldschermen dat nodig is om in 95% van de gevallen de transakties per dag te kunnen verwerken, moet bij de T.T.T. worden opgeteld: 1.64 x de standaard afwijking (17). De standaardafwijking wordt door het programma berekend en afgedrukt. Zie Fig. 44.5. Op basis daarvan worden de nieuwe beeldschermuren per dag berekend en de rest van het rekenproces verloopt als hier boven is aangegeven.
- Het is duidelijk dat op deze wijze ook pieksituaties snel doorgerekend kunnen worden. Wanneer het gaat om een dagpiek die niet langer dan ca. 1 uur duurt, moet worden uitgegaan van de netto T.T.T., omdat in zo'n periode in het algemeen pauzes en persoonlijke verzorging komen te vervallen.
- Veel computersystemen zijn volgens de ontwerpers overgedimensioneerd uit veiligheidsoverwegingen. Soms worden zelfs percentages genoemd. Zolang echter niet is vastgesteld waar de 100% betrekking op heeft, zeggen deze percentages natuurlijk niets. Voor het aantal beeldschermen staat de 100% nu vast, ofwel voor het gemiddelde aantal, ofwel voor 95% van de gevallen ofwel in een pieksituatie. Wanneer het aantal beeldschermuren is bepaald, is bij een achturige werkdag het aantal beeldschermen te bepalen.

Wanneer in WPGB 3 beeldschermen worden genomen dan is niet exakt voldaan aan de eis dat het werk voor 12.00 uur gedaan is. De vraag is natuurlijk hoe nauwkeurig het getal 12.31 is. De T.T.T. wordt berekend in twee cijfers achter de komma. Dit suggereert een nauwkeurigheid die er in werkelijkheid niet is. Wanneer er de T.T.T. voor de helft bestaat uit een ruw geschatte aan- en uitloop, dan zeggen cijfers achter de komma niets. In de praktijk is gebleken dat een nauwkeurigheid van 10% haalbaar is. Wanneer dialoogsimulatie uitgevoerd is, kan de T.T.T. heel goed gekontroleerd worden met een meting. In het voorbeeld zijn dus drie beeldschermen genomen, met daarbij de kanttekening dat daar geen overkapaciteit in zit. Bij WPGF en WPGG is nog wat reserve aanwezig.

- Bij WPGD zit het precies goed, maar de drie medewerkers doen dan ook de hele dag niets anders dan transaktie T5 uitvoeren. Als afdeling D uit drie medewerkers bestaat dan zou er nu wel eens een probleem kunnen zijn ontstaan. Laten we eens aannemen dat de gebruikers, gezien de sociale aspekten, die nu in cijfers zijn weergegeven, deze situatie niet akseptabel vinden, omdat er op de afdeling meer te doen is dan alleen maar inkloppen. Alvorens te besluiten 4 beeldschermen te nemen en een personeelsadvertentie te plaatsen voor een vierde medewerker, moet het volgende gekontroleerd worden.
- Hoe nauwkeurig is het aantal transakties per dag? Als de gebruiker tijdens het onderzoek een ruwe schatting heeft gedaan, dan is hij nu zeker genegen er nog eens wat nauwkeuriger naar te kijken.
- Waar zitten onnauwkeurigheden in het detailschema, die de T.T.T. in belangrijke mate beinvloeden? Als het gaat om schattingen van de gebruiker geldt hetzelfde als voor het aantal tranakties per dag. Als het gaat om nieuwe aspekten die niemand nog uit ervaring kent, is een meting nu zeker op z'n plaats. Voor zo'n meting moet een echte werkomstandigheid opgebouwd worden en de dialoogsimulator fungeert als beeldscherm.
- Hoe is de compositie van de terminaltransaktietijd en welke elementen maken het leeuwendeel uit? Valt er nog iets te verbeteren aan de hele procedure?
- Is er met een heel ander transaktie-ontwerp nog een verbetering te bereiken? De gebruiker staat voor de keus: een extra medewerker of een andere transaktie. Hij is nu in ieder geval gemotiveerd een aantal alternatieven in aanmerking te nemen.

En wanneer dat allemaal niet helpt mag de gebruiker natuurlijk in overleg met z'n medewerkers best besluiten om te beginnen met drie beeldschermen omdat het aantal transakties per dag misschien toch niet zo konstant is. In ieder geval weet nu iedereen wat 100% is en dat de overkapaciteit 0% is.

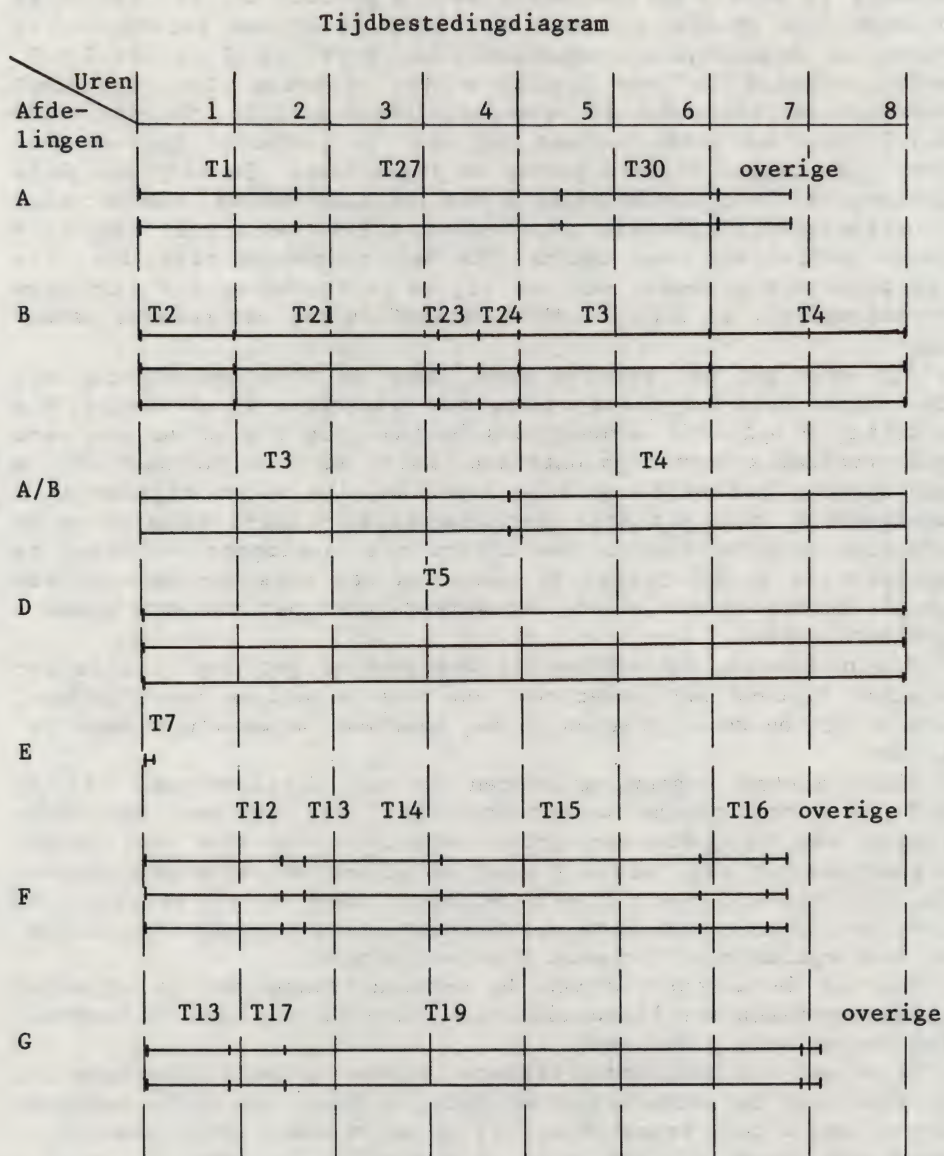


Fig. 44.7 Tijdverdeling per werkplek

- Op de afdeling E hebben de gebruiker in ieder geval een transaktie in het ontwerp ingebracht. Hoewel de terminal nauwelijks bezet zal worden, kunnen er natuurlijk allerlei politieke redenen bestaan om er toch maar een beeldscherm neer te zetten. Het kan ook zijn dat er binnen een ander projekt meer transakties voor deze afdeling zullen ontstaan en dat er daarom maar vast een scherm geplaatst wordt.
- De transakties T8, T9, T10, T11, T17, T18 en T19 zijn transakties die elders al in bedrijf zijn. Dus werd de netto T.T.T. gemeten, omgerekend naar de bruto-T.T.T. en zo in de tabel opgenomen.
- We gaan nog even terug naar WPGD. Met drie beeldschermen zouden alle transakties iedere dag precies uitgevoerd kunnen worden. Stel nu, dat het gaat om een afdeling van vijf medewerkers en dat de afdelingschef het nodig vindt dat er bij ieder bureau een beeldscherm komt. De informatie-analist zal er geen problemen mee hebben. Gemiddeld over een dag maakt het voor het systeem niet uit met hoeveel beeldschermen een bepaald aantal transakties wordt uitgevoerd, als we even aannemen dat de overhead voor een aangelogde, doch niet aktieve terminal verwaarloosbaar is. Bij de bepaling van de systeembelasting moet wel degelijk rekening gehouden worden met de situatie dat de vijf beeldschermen alle in gebruik zijn. Met andere woorden, hoewel de uitbreiding van het aantal beeldschermen bij een gegeven aantal transakties ongevaarlijk lijkt, zal het bijdragen tot het ontstaan van pieken in de systeembelasting. In het deel voor de transaktie-analist wordt dit verder uitgewerkt.
- De presentatie van de resultaten aan de gebruikers kan op allerlei manieren plaatsvinden. In Fig. 44.7 is een voorbeeld gegeven van een lijnendiagram waarin is uitgegaan van een gelijkmatige verdeling van transakties over de beeldschermen van een groep. De gebruiker mag elke andere verdeling aangeven.
- Wanneer de informatie-analist beschikt over een P.C. dan kan hij met MULTIPLAN-achtige pakketten overzichten als Fig. 44.6 snel maken en nog sneller allerlei situaties doorrekenen.
- Dat laatste gaat nog meer spreken wanneer we ook de logische en technische resultaten erbij betrekken. Dan ontstaan nog veel meer kolommen en is een P.C. een uitstekend hulpmiddel. Wanneer MULTIPLAN-achtige pakketten nog eens voor mainframes beschikbaar komen, kunnen we de resultaten via de datadictionary voor informatie-analisten, transaktie-analisten en gebruikers op elk beeldscherm snel beschikbaar hebben.

Voorbeeld 2. Bepaling van de wachtrijlengte voor het loket.
In dit voorbeeld gaat het om ergonomische Transaktie analyse. Een bedrijf gaat een handmatige procedure aan loketten vervangen

Terminal Transaktie tijd (T.T.T.)

Rubriek	Tussen-waarden		Aantal seconden	Variantie
Invoerlengte	15.80			
Intiksnelheid	3.00			
	--------	/		
Invoertijd			5.27	3.60
Wachttijden en denktijden			14.80	30.22
Responses	2.05			
Transportvertraging	0.00			
	--------	*		
Transporttijd			0.00	0.00
Response-eenheden	2.05			
Tijd per eenheid	0.05			
	--------	*		
Verwerkingstijd			0.10	0.00
Uitvoerlengte	13.00			
Afdruksnelheid	2.50			
	--------	/		
Uitvoertijd			5.20	0.16
			----------	----------
Subtotaal			25.37	33.98
Foutherstelpercentage	5.00			
Foutherstel-tijd			1.27	0.16
Aanloop en uitloop			2.50	0.60
			----------	----------
Netto T.T.T.			29.14	34.82
(Volledig effektief)	Standaardafwijking			5.90
Bruto T.T.T.			41.63	106.43
(Bij 70% effektief)	Standaardafwijking			10.32

Procentuele verdeling van de T.T.T.

Invoertijd	18.08 %
Wachttijden en denktijden	50.79 %
Dialoogresponse-tijden	0.18 %
Afsluitresponse-tijden	0.17 %
Uitvoer-tijd	17.86 %
Tijd voor fout herstellen	4.76 %
Tijd voor aanloop en uitloop	8.58 %

Fig. 44.8 De T.T.T.-resultaten van de lokettransaktie.

door een transaktie. Men wil zeker tijdens het logisch ontwerp inzicht hebben in de gemiddelde wachtrijlengte in de normale situatie en in het aantal benodigde loketten om de pieken op te vangen. Pieken treden 10 tot 15 keer per dag op: er lopen dan ineens 20 tot 30 mensen binnen. Voor de normale situatie gaan we uit van een Poisson-aankomstpatroon, omdat er anders helemaal niets te rekenen valt. Een gemiddelde waarde voor het aantal mensen per uur in de normale situatie is moeilijk aan te geven, dus wil de gebruiker een aantal situaties berekend hebben. Transaktie analyse wordt uitgevoerd en de resultaten zijn vrij nauwkeurig het om een simpele transaktie gaat. Fig. 44.8 geeft de T.T.T. In deze situatie heeft het natuurlijk geen zin om met de bruto T.T.T. te werken, omdat pauzes en persoonlijke verzorging in dit soort omgevingen anders zijn geregeld: klanten worden altijd direkt geholpen en koffiedrinken gebeurt in shifts of op de momenten dat er geen klanten zijn. We gaan er vanuit dat de wachtrijtheorie, zoals die behandeld wordt in (17), bekend is. De netto T.T.T. is de servicetijd. In de wachtrijtheorie wordt de bezetting van de server, in dit geval de loketbediende, berekend en op basis daarvan wordt de wachtrijlengte bepaald.
De bezetting van de server, B = E(n) x E(ts), waarin
- B: de bezetting
- E(n): gemiddeld aantal aankomsten per tijdseenheid
- E(ts): de gemiddelde terminaltransaktietijd: 29.14 sec.

Gem.aantal aankomsten per minuut	Aantal loketten	Bezetting per loket	Kan dat alle loketten bezig zijn	Gem.lengte van de wachtrij per loket
1	1	(1/60)x29.14 =0.48	0.48	0.95
2	1	(2/60)x29.14/ =0.97	0.97	veel
1	2	(1/60)x29.14/2=0.24	0.1	0.2
2	2	(2/60)x29.14/2=0.48	0.32	0.5
3	2	(3/60)x29.14/2=0.72	0.6	1.3
4	2	(4/60)x29.14/2=0.97	0.95	veel
1	3	(1/60)x29.14/3=0.16	0.02	0.2
2	3	(2/60)x29.14/3=0.32	0.09	0.3

Fig. 44.9 Verloop van de normale situatie.

Grootte van de piek (aantal personen)	Aantal loketten	Gemiddelde doorlooptijd in minuten
20	1	9.7
25	1	12.14
30	1	14.57
20	2	4.85
25	2	6.07
30	2	7.28
20	3	3.2
25	3	4.04
30	3	4.85

Fig. 44.10 Doorlooptijd van pieken.

Op basis van de grafieken (8 of 44) kan het overzicht van Fig. 44.9 worden samengesteld. De laatste kolom levert eigenlijk weinig informatie. Dat wordt veroorzaakt door het feit dat boven een belasting van 0,8 de grafieken zo stijl verlopen dat geen bruikbaar cijfer kan worden afgelezen.
Voor het verloop van de pieksituatie kan een tabel worden gemaakt als is weergegeven in Fig. 44.10. De berekeningen zijn niet spectaculair, maar illustreren dat allerlei situaties nu kunnen worden doorgerekend. Boeken over de wachtrijtheorie staan meestal vol voorbeelden. Met voorbeeld 2 is duidelijk gemaakt dat bij interaktieve toepassingen die berekeningen gemaakt kunnen worden omdat Transaktie analyse de cijfers levert. Ook ingewikkelde situaties met verschillende transakties en loketten van bepaalde transakties kunnen nu naar alle gewenste gezichtspunten worden doorgerekend.

Voorbeeld 3. Is er voldoende tijd beschikbaar?
In dit voorbeeld gaat het om een globale logische Transaktie analyse. Er zijn ruwe schetsen van gegevens- en funktiemodellen. Er is nog niets gedaan aan transaktie-ontwerp of beeldschermontwerp. Wel is er een aantal interaktieve toepassingen bedacht, zij het in grote lijnen. De vragen die langzamerhand boven beginnen te komen betreffen de timing van het geheel. Hoeveel uren zit iemand achter een beeldscherm, komt hij nog toe aan andere taken, hoe-

veel printkapaciteit is er nodig per beeldschermgroep, hoe groot is de systeembelasting per werkplek?
Er worden door de informatie-analist in twee dagen tijd 30 globale transaktieschema's gemaakt, uiteraard niet in overleg met de gebruiker. In de eerste plaats omdat de gebruiker nog niet betrokken was bij het ontwerp en in de tweede plaats omdat het alleen ging om een indruk te krijgen van tijdsaspekten van de transakties. Wanneer transaktieschema's tijdens het logisch ontwerp gemaakt worden samen met de gebruikers, ligt het tempo wat lager dan 15 per dag.
Uiteraard kunnen we niet van alle transakties de transaktieschema's, detailschema's en resultaten afdrukken. Om een indruk te geven, nemen we de tijdsbesteding op de werkplek van de afdelingschef. Voor het zijn de volgende transakties bedacht A18, A37, A17, B39, B32 en A32. Deze laatste betreft het beoordelen van een ordervoorstel zoals dat door het systeem is opgesteld. Dat gebeurt maar eenmaal per dag. Om te voorkomen dat die transaktie het gewogen gemiddelde van de combitransaktie te sterk beinvloed, zijn de resultaten van A32 achteraf bij die van de combitransaktie geteld. Zie Fig. 44.17 en 44.19.
In Fig. 44.11 is het globale transaktieschema weergegeven. In Fig. 44.12 - Fig. 44.16 is de komplete output van het rekenprogramma weergegeven: eerst twee pagina's met het ingevoerde detailschema vervolgens de drie pagina's met resultaten. Voor de overige transakties is ook Transaktie analyse uitgevoerd. Om een indruk te krijgen van het geheel, is een combitransaktie gemaakt. Het detailschema bestaat uit de cijfers van de parameteroverzichten van de samenstellende transakties, en kansfaktoren die zijn berekend uit het aantal transakties per dag. De berekening van de kansfaktoren is als kommentaar opgenomen in het detailschema. In Fig. 44.17 is het eerste blad van het detailschema weergegeven, in Fig. 44.18 het tweede en in Fig. 44.19 het laatste. De rest van het detailschema kan men zich nu wel voorstellen. De resultaten van het rekenprogramma zijn weergegeven in Fig. 44.20-44.22. Hoe de resultaten zijn verwerkt is achteraf, als kommentaar opgenomen in het detailschema, zie Fig. 44.19.
Een reeks '*' in de kolom 'VARIANTIE' betekent dat de variantie een te grote waarde heeft om afgedrukt te worden. In een globale Transaktie analyse wordt meestal alleen gewerkt met het gemiddelde. De variantie ontstaat echter zo gauw ergens een kansfaktor wordt gebruikt die kleiner is dan 1.
De konklusie voor de werkplek van de afdelingschef is dus opzienbarend: 23.1 + 2.2 uren per dag achter het beeldscherm! (Fig. 44.19) Natuurlijk gaat het om globale cijfers, maar het is iedereen duidelijk dat hier iets anders aan de hand is dan de nauwkeurigheid van de cijfers. De hele opzet van de transakties moet

worden herzien.
In dit voorbeeld is door het werken met een combitransaktie een gemiddelde transaktie ontstaan voor de werkplek van de afdelingschef. In een globaal onderzoek is dat vaak voldoende nauwkeurig. De werkwijze in voorbeeld 1 heeft het voordeel dat men zicht blijft houden op de verschillende transakties van een werkplek. De aanpak hangt af van de konklusies die men wil trekken.

Transaktieschema centraal

Transaktienaam: A32 behandeling ordervoorstel

Menselijke handelingen	Transport	Machinale verwerking
Intypen keuze via menuscherm	----)	Raadplegen Order voorstellen, bestand, artikelbestand, leveranciersbestand. Selektie scherm + vullen
lezen en interpreteren scherm 1 t/m N.	(----	informatie.
Indien details gewenst Opvragen van deze informatie	----)	Raadplegen kleuren/maten best. stock bestand. order/order line best. mail-plan best. Opbouw aanvullende infor-
Lezen interpreteren	(----	matie (zelfde scherm).
Indien nodig: Wijzigen ordervoorstel t.a.v. hoeveelheid, afleverdatum, additionele informatie	----)	Kontrole en Bijwerken Ordervoorstellen best. order/order line best.
Indien gewenst artikelen toevoegen aan ordervoor- stel.	(----	
Artikelnr intypen	----)	Opzoeken artikelgegevens Berekenen ordervoorst.
lezen en interpreteren inkl.	(----	
detailgegevens.	----)	Kontrole en bijwerken ordervoorstellen order/order line
Wanneer alle schermen verwerkt en goedgekeurd	(----	"OK"
zijn start batch orders	----)	Schoon keuze scherm
	(----	

Fig. 44.11 Globaal transaktieschema.

A32 Behandeling ordervoorstel

Kommentaar	Kode	Kans-faktor	Gemid. waarde	Variantie
Intiksnelheid	13	1.000	1.00	0.0
Transportvertraging	21	1.000	1.00	0.0
Tijd per response-eenheid	22	1.000	0.10	0.0
Afdruksnelheid n.v.t.				
Intoetsen keuze via menuscherm	1	1.000	1.00	0.0
Transport heen	2	1.000	4.00	0.0
Enter	5	1.000	1.00	0.0
Selektie masker ordervoorstel	6	1.000	1.00	0.0
Transport terug	3	1.000	750.00	0.0
Voor 100 schermen: (5 art.p.sch.)				
Denk en wachtt. (10 sec.p.scherm)	12	1.000	1000.00	500.0
Intypen enter	1	1.000	1.00	0.0
Transport heen	2	1.000	1.00	0.0
Enter	5	1.000	1.00	0.0
Schrijven spool/order/order line	6	1.000	500.00	100000.0
Raadplegen ordervoorstel	6	1.000	500.00	100000.0
artikel	6	1.000	1000.00	200000.0
leverancier	6	1.000	50.00	100.0
Transport artikelinfo (5x100x100)	3	1.000	50000.00	1000.0
Bij einde (= laatste scherm):				
Intypen enter	1	1.000	1.00	0.0
Transport heen	2	1.000	1.00	0.0
Enter	7	1.000	1.00	0.0
Bijwerken spool order/orderline	8	1.000	1.00	0.0
Selektie keuze masker	8	1.000	1.00	0.0
Transport terug	3	1.000	100.00	0.0
Bij opvragen detailgegevens:				
D. en wachtt.(extr 10 sec.p.sch)	12	0.200	5000.00	500.0
	****	1.000	1000.00	********
Intoetsen regelnummer	1	0.200	1000.00	0.0
	****	1.000	200.00	159999.94
Transport heen	2	0.200	2500.00	0.0
	****	1.000	500.00	999999.25
Enter	5	0.200	500.00	0.0
	****	1.000	100.00	40000.0

Fig. 44.12 Eerste deel van het detailschema.

A32 Behandeling ordervoorstel

Kommentaar	Kode	Kans-faktor	Gemid. waarde	Variantie
Raadplegen kleuren/maten	6	0.200	500.00	10000.0
	****	1.000	100.00	42000.0
stock	6	0.200	500.00	10000.0
	****	1.000	100.00	42000.00
order/orderline	6	0.200	500.00	10000.00
	****	1.000	100.00	42000.00
mail plan	6	0.200	500.00	10000.00
	****	1.000	100.00	42000.00
Transport terug	3	0.200	150000.00	10000.00
	****	1.000	30000.00	********
Bij wijzigingen:				
Hoeveelheid/afleverdat/addit.info	1	0.200	10000.00	100.00
	****	1.000	2000.00	********
Transport heen	2	0.200	12500.00	100.00
	****	1.000	2500.00	********
Enter	5	0.200	500.00	0.0
	****	1.000	100.00	40000.00
Bijwerken ordervoorstel	6	0.200	500.00	10000.00
	****	1.000	100.00	42000.00
order/orderline	6	0.200	500.00	10000.00
	****	1.000	100.00	42000.00
Bij toevoegingen:				
Intoets.toevoeg.+artnr(30 art)	1	1.000	390.00	100.00
Transport heen	2	1.000	1000.00	100.00
Enter	5	1.000	30.00	100.00
Raadpleeg artikel	6	1.000	30.00	100.00
leveranciers	6	1.000	30.00	100.00
Display masker	3	1.000	750.00	100.00
display info (100 + 300)	3	1.000	400.00	100.00
Lezen + beoordelen	11	1.000	60.00	100.00
Intoetsen wijzigingen (20x30)	1	0.200	600.00	100.00
	****	1.000	120.00	57619.94
Enter	2	1.000	30.00	100.00
Bijwerken spool file	6	1.000	30.00	100.00
ordervoorstel	6	1.000	30.00	100.00
order/orderline	6	1.000	30.00	100.00
Masker clearen	3	1.000	150.00	0.0
Einde invoer	99			

Fig. 44.13 Tweede deel van het detailschema

A32 Behandeling ordervoorstel

Parameter-overzicht

Kode rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:			
1 Invoerlengte	2713.00	aanslagen	********
2 Transportlengte invoer	4036.00	tekens	********
3 Transportlengte uitvoer	82149.94	tekens	********
4 Uitvoerlengte	0.0	tekens	0.00
5 Dialoogresponses	232.00	responses	80100.00
6 Dialoogsresponse-eenheden	2852.00	eenheden	341109.50
7 Afsluitresponses	1.00	responses	0.00
8 Afsluitresponse-eenheden	2.00	eenheden	0.00
Personele-parameters:			
11 Aanloop en uitloop	62.50	seconden	100.60
12 Wachttijden en denktijden	2000.00	seconden	********
13 Intiksnelheid	1.00	tek./sec.	0.00
14 Foutherstelpercentage	5.00	procent	2.40
15 Min. invoer repetitietijd	0.00	seconden	0.00
16 Effektieve werktijd	70.00	procent	100.00
Machine-parameters:			
21 Transportvertraging	1.00	seconden	0.00
22 Tijd per response-eenheid	0.10	seconden	0.00
23 Afdruksnelheid	0.00	tek./sec.	0.00

Fig. 44.14 Het parameteroverzicht.

A32 Behandeling ordervoorstel

Rubriek	Tussen-waarden		Aantal seconden	Variantie
		Terminal Transaktie tijd (T.T.T.)		
Invoerlengte	2713.00			
Intiksnelheid	1.00			
	--------	/		
Invoertijd			2713.00	********
Wachttijden en denktijden			2000.00	********
Responses	233.00			
Transportvertraging	1.00			
	--------	*		
Transporttijd			233.00	80099.94
Response-eenheden	2854.00			
Tijd per eenheid	0.10			
	--------	*		
Verwerkingstijd			285.40	3411.09
			----------	----------
Subtotaal			5231.39	********
Foutherstelpercentage	5.00			
Foutherersteltijd			261.57	57322.66
Aanloop en uitloop			62.50	100.60
			----------	----------
Netto T.T.T.			5555.46	********
(Volledig effektief)			Standaardafwijking	4512.12
Bruto T.T.T.			7936.37	********
(Bij 70% effektief)			Standaardafwijking	6544.84

Procentuele verdeling van de T.T.T.

Invoertijd	48.83 %
Wachttijden en denktijden	36.00 %
Dialoogresponse-tijden	9.31 %
Afsluitresponse-tijden	0.02 %
Uitvoer-tijd	0.00 %
Tijd voor fout herstellen	4.76 %
Tijd voor aanloop en uitloop	1.13 %

Fig. 44.15 De pagina: Terminaltransaktietijd.

A32 Behandeling ordervoorstel

Rubriek	Gemiddelde waarde		Variantie
Lijn- en responsetijd aspekten			
Invoerrepetitietijd ongunstig	23.84 seconden		1213.79
Invoerrepetitietijd normaal	34.06 seconden		2500.81
Gem.berichtlengte invoer	17.32 tekens		921.82
Gem.berichtlengte uitvoer	354.57 tekens		249721.44
Totaal per response	369.89 tekens		250643.00
Gem. tijd per dialoogresponse	2.23 seconden		2.31
Gem. tijd per afsluitresponse	1.20 seconden		0.00
Gem. tijd per response	2.22 seconden		2.28
Gemiddelde verwerkingstijd			
per dialoogresponse	1.23 seconden		2.31
per afsluitresponse	0.20 seconden		0.00
per response	1.22 seconden		2.28
Responsetijd korter dan in X%	99%	95%	90%
Dialoogresponse	5.77	4.72	4.18
Afsluitresponse	1.20	1.20	1.20
Gemiddelde response	5.74	4.70	4.16

Fig. 44.16 De pagina: Lijn- en response-aspekten.

Combitransaktie afd.chef

Kommentaar

Werkplek: afd.chef

Transakties met
aantal en percentages:

Tr.akt	aantal	%
A18	50	0.047
A37	30	0.027
A17	850	0.787
B39	50	0.047
B32	100	0.092

Totaal	1080	

A32 niet meegenomen,
achteraf bijgeteld.

Voor alle transakties geldt:

	Kode	Kans-faktor	Gemidd. waarde	Variantie
13 Intiksnelheid	13	1.000	1.00	0.00
14 Foutherstelpercentage				
15 Min. invoer repetitietijd	15	1.000	0.00	0.00
16 Effektieve werktijd				
21 Transportvertraging	21	1.000	1.00	0.00
22 Tijd per response eenheid	22	1.000	0.10	0.00
23 Afdruksnelheid A17	-23	0.338	180.00	0.00
,, B32		0.662	600.00	0.00
		1.000	458.04	39740.69

Fig. 44.17 Eerste blad van het detailschema.

Combitransaktie afd.chef

Kommentaar		Kode	Kans-faktor	Gemid. waarde	Variantie
A18 Inhouse present.					
1	Invoerlengte	1	0.027	40.00	0.00
		****	1.000	1.08	42.03
2	Transportlengte invoer	2	0.027	52.00	0.00
		****	1.000	1.40	72.04
3	Transportlengte uitvoer	3	0.027	2250.00	200.00
		****	1.000	60.75	133002.31
4	Uitvoerlengte	4	0.027	0.00	0.00
		****	1.000	0.00	0.00
5	Dialoogresponses	5	0.027	2.00	0.00
		****	1.000	0.05	0.11
6	Dialoogresponse-eenheden	6	0.027	22.00	0.00
		****	1.000	0.59	12.72
7	Afsluitresponses	7	0.027	0.00	0.00
		****	1.000	0.00	0.00
8	Afsluitresponse-eenheden	8	0.027	0.00	0.00
		****	1.000	0.00	0.00
11	Aanloop en uitloop	11	0.027	2.50	0.60
		****	1.000	0.07	0.18
12	Wachttijden en denktijden	12	0.027	0.00	0.00
		****	1.000	0.00	0.00
A37 Stock Man.inq.					
1	Invoerlengte	1	0.787	40.00	0.00
		****	1.000	31.48	268.21
2	Transportlengte invoer	2	0.787	52.00	0.00
		****	1.000	40.92	453.27
3	Transportlengte uitvoer	3	0.787	1750.00	100.00
		****	1.000	1377.25	513450.00
4	Uitvoerlengte	4	0.787	0.00	0.00
		****	1.000	0.00	0.00
5	Dialoogresponses	5	0.787	2.00	0.00
		****	1.000	1.57	0.67
6	Dialoogresponse-eenheden	6	0.787	5.00	4.00
		****	1.000	3.93	7.34

Fig. 44.18 Het tweede blad van het detailschema.

Combitransaktie afd.chef

Kommentaar	Kode	Kans-faktor	Gemid. waarde	Variantie

Berekeningen:

Bruto T.T.T.: 77 sec.

$$\frac{1080 \times 77}{3600} = 23.1 \text{ uur}$$

Verkeer: 978 tekens in 26 sec.
d.w.z. 37 tek/sec.

CPU belasting:

34 file access. (F.A.) in 77 sec.
d.w.z. 0.4 F.A./sec.

Ordervoorstel bruto T.T.T.: (A32)

$$\frac{8000}{3600} = 2.2 \text{ uur}$$

Verkeer:
370 tekens in 24 sec.
d.w.z. 15 tek./sec.
CPU belasting:
2854 file access. in 8000 sec.
d.w.z. 0.3 F.A./sec.

Einde invoer	99			

Fig. 44.19 Het laatste blad van het detailschema.

Combitransaktie afd. chef

Parameter-overzicht

Kode rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:			
1 Invoerlengte	41.31	aanslagen	979.64
2 Transportlengte invoer	53.86	tekens	1586.14
3 Transportlengte uitvoer	2002.10	tekens	*********
4 Uitvoerlengte	74.60	tekens	132440.19
5 Dialoogresponses	2.10	responses	1.93
6 Dialoogresponse-eenheden	33.89	eenheden	63474.44
7 Afsluitresponses	0.00	responses	0.00
8 Afsluitresponse-eenheden	0.00	eenheden	0.00
Personele-parameters:			
11 Aanloop en uitloop	2.50	seconden	2.89
12 Wachttijden en denktijden	1.85	seconden	24.19
13 Intiksnelheid	1.00	tek./sec.	0.00
14 Foutherstelpercentage	5.00	procent	2.40
15 Min. invoer repetitietijd	0.00	seconden	0.00
16 Effektieve werktijd	70.00	procent	100.00
Machine-parameters:			
21 Transportvertraging	1.00	seconden	0.00
22 Tijd per response-eenheid	0.10	seconden	0.00
23 afdruksnelheid	458.04	tek./sec.	39740.69

Fig. 44.20 Het parameterovericht.

Combitransaktie afd. chef

Rubriek	Tussen-waarden		Aantal seconden	Variantie
Terminal Transaktie tijd (T.T.T.)				
Invoerlengte	41.31			
Intiksnelheid	1.00			
	--------	/		
Invoertijd			41.31	979.64
Wachttijden en denktijden			1.85	24.19
Responses	2.10			
Transportvertraging	1.00			
	--------	*		
Transporttijd			2.10	1.93
Response-eenheden	33.89			
Tijd per eenheid	0.10			
	--------	*		
Verwerkingstijd			3.39	634.74
Uitvoerlengte	74.60			
Afdruksnelheid	458.04			
	--------	/		
Uitvoertijd			0.16	0.64
Subtotaal			48.82	1641.15
Foutherstelpercentage	5.00			
Foutherstel tijd			2.44	4.67
Aanloop en uitloop			2.50	2.89
Netto T.T.T.			53.76	1648.72
(Volledig effektief)	Standaardafwijking			40.60
Bruto T.T.T.			76.39	3485.08
(Bij 70% effektief)	Standaardafwijking			59.03

Procentuele verdeling van de T.T.T.

Invoertijd	76.85 %
Wachttijden en denktijden	3.44 %
Dialoogresponse-tijden	10.21 %
Afsluitresponse-tijden	0.00 %
Uitvoer-tijd	0.30 %
Tijd voor fout herstellen	4.76 %
Tijd voor aanloop en uitloop	4.65 %

Fig. 44.21 De pagina: Terminaltransaktietijd.

Combitransaktie afd. chef

Rubriek	Gemiddelde waarde		Variantie
Lijn- en responsetijd aspekten			
Invoerrepetitietijd ongunstig	28.58 seconden		659.51
Invoerrepetitietijd normaal	36.54 seconden		1373.19
Gem.berichtlengte invoer	25.63 tekens		646.39
Gem.berichtlengte uitvoer	952.75 tekens		********
Totaal per response	978.38 tekens		********
Gem. tijd per dialoogresponse	2.61 seconden		144.88
Gem. tijd per afsluitresponse	0.00 seconden		0.00
Gem. tijd per response	2.61 seconden		144.88
Gemiddelde verwerkingstijd			
per dialoogresponse	1.61 seconden		144.88
per afsluitresponse	0.00 seconden		0.00
per response	1.66 seconden		144.88
Responsetijd korter dan in X%	99%	95%	90%
Dialoogresponse	30.66	22.35	18.02
Afsluitresponse	0.00	0.00	0.00
Gemiddelde response	30.66	22.35	18.02

Fig. 44.22 De pagina: Lijn- en responsetijdaspekten.

Voorbeeld 4. Even wachten op de faktuur.
Dit voorbeeld betreft de ergonomische resultaten van een technische Transaktie analyse. Bij de balieverkoop van onderdelen worden beeldschermen ingezet voor het invoeren van orders en een printer voor het printen van de fakturen. Het projekt bevindt zich gedeeltelijk al in de bouwfase als iemand zich zorgen begint te maken over de kapaciteit van het kleine printertje voor de fakturen. Het systeem is een eenvoudige minicomputer die ook nog voor wat andere toepassingen wordt gebruikt. De eenvoud blijkt te liggen in het beperkte aantal schijven, het zeer beperkte aantal files dat tegelijkertijd geopend mag zijn en een beperkt geheugen met veel I/O's voor het laden van programma's en het openen van bestanden. Ten dienste van het order entry-programma draait er voor alle beeldschermen een scan-programma en als er geprint moet worden een printprogramma dat gebruik maakt van de spool-file. Het scan-programma werkt bestanden bij en bereidt het printen voor. Dit programma verwerkt iedere ingevoerde order. Een analyse van de systeemstroomschema's van het programma toont aan dat het per transaktie van zes orderregels ca 1100 diskaccessen uitvoert. Het printprogramma heeft 150 diskaccessen nodig bij zes orderregels per order. Op een leegstaande machine is het printprogramma al getest en het printen van een faktuur met 6 faktuurregels duurt ongeveer 30 seconden. Als er ook andere programma's aktief zijn blijkt het 60 seconden te gaan duren. De enige mogelijkheid om er nog wat aan te doen blijkt een snellere printer. Er wordt besloten Transaktie analyse uit te voeren om de zaak in kaart te brengen. In Fig. 44.23 - 44.26 is de analyse weergegeven.
Een transaktie duurt gemiddeld 228 seconden en er worden 339 diskaccessen uitgevoerd. De cijfers voor denk- en wachttijden zijn zeker niet aan de lage kant. De klant weet namelijk nooit een artikelnummer en de baliebediende kent er maar een beperkt aantal uit z'n hoofd. Voor zes artikelen is 60 seconden een redelijk cijfer. Een diskaccess duurt volgens de gegevens van de leverancier gemiddeld 0.05 seconden. Dat betekent dat het scanprogramma minstens 1100 x 0.05 = 55 seconden duurt. Dus na het invoeren van een transaktie begint na 55 seconden de printer met het afdrukken van de faktuur.
Het scan-programma draait dan kontinu omdat in 228 seconden 5.1 x 1100 diskaccessen kunnen worden uitgevoerd. In 228 seconden kunnen 228/30 = 7.6 fakturen worden geprint, dus het printprogramma is bij zes beeldschermen bijna konstant aktief. Het lijkt allemaal net te kunnen totdat we de schijfbelasting bij zes beeldschermen in rekening brengen: B = E(n) x E(ts) = ((6x339 + 5.1 x 1100 + 6x150)/228)x0.05 = 1.8! Bij twee beeldschermen is de belasting al ((2x339 + 2x1100 + 2x 150)/228)x0.05 = 0.69. Volgens

de wachtrijtheorie duurt een diskaccess bij deze belasting al twee keer zo lang als normaal. Het duurt dus nu al bijna twee minuten (2 x 55 = 110 sec.) om van een transaktie de bestanden bij te werken en het printen voor te bereiden. Het printen zelf duurt normaal 150x0.05 + 22.5 = 30 seconden. Dat wordt nu 37.5 seconden. De terminaltransaktietijd bestaat slechts voor 7.56% uit verwerkingstijd, zie Fig. 44.25. Dat betekent dat de responsetijden 2 x zo lang zullen zijn als berekend, maar dat de T.T.T. slechts weinig zal toenemen. Per uur kunnen bij twee beeldschermen dus nog steeds ca. 2 x 3600/228 = 32 orders worden ingevoerd. Er kunnen echter maar (3600-32x110)/37.5= 2,1 fakturen in een uur worden geprint en dat is erg weinig.
Bij 3 beeldschermen is de schijfbelasting al 1 en zijn de wachttijden al oneindig hoog.
Wat begon als een onderzoek naar de benodigde printkapaciteit, eindigt in een analyse van de hoeveelheid I/O's. Het scan-programma blijkt het knelpunt te zijn in het geheel. Dan blijkt dat er slechts twee files tegelijk geopend mogen zijn. Daardoor staan de programma's vol met OPEN- en CLOSE-opdrachten. Iedere OPEN blijkt een tiental I/O's te kosten omdat de volume table of contents van de schijf sequentieel gelezen wordt. Tenslotte wordt duidelijk dat het hele koncept van de software voor deze machine ongeschikt is: veel te veel stuur- en hulpbestanden.
Er wordt een hoeveelheid verwerking naar de batch verschoven, want achteraf blijkt dat de gebruiker het niet perse noodzakelijk vindt dat alle bestanden iedere seconde up-to-date zijn. En zo blijkt ook in dit projekt, dat zou zijn vastgelopen in de performance-problemen, dat er fouten in de analysefase zijn gemaakt. Gelukkig dat men er nu tijdens het technisch ontwerp nog achter kwam.

BALIEVERKOOP

Kommentaar	Kode	Kans-faktor	Gemid. waarde	Variantie
Tweede situatie				
Intiksnelheid	13	1.000	1.50	1.00
Transport vertraging	21	1.000	0.01	0.00
Afdruksnelheid	23	1.000	999.00	0.00
Tijd per resp. eenh.	22	1.000	0.05	0.01
*				
Start van transaktie				
*				
Naam klant vragen	11	1.000	5.00	4.00
Zoekargument invoeren	1	1.000	6.00	0.00
Dialoogresponse	5	1.000	1.00	0.00
Aantal eenheden	6	1.000	6.00	2.00
*				
Kop				
*				
Openen van bestanden in MO	6	1.000	97.00	0.00
Intoetsen klantnr.	1	1.000	8.00	0.00
Dialoogresponse	5	1.000	1.00	0.00
Aantal eenheden	6	1.000	3.00	0.00
Intoetsen 'BALIE VERKOOP'	1	1.000	1.00	0.00
Intoetsen mutatie kode	1	1.000	2.00	0.00
Intoetsen 'ACC'	1	1.000	2.00	0.00
Dialoogresponse	5	1.000	3.00	0.00
Intoetsen funktie	1	1.000	1.00	0.00
Dialoogresponse	5	1.000	1.00	0.00
Aantal eenheden	6	1.000	10.00	0.00
Intoetsen klanten ref.	1	1.000	10.00	2.00
Intoetsen 'BESTELD DOOR'	1	1.000	8.00	5.00
Intoetsen 'ACC'	1	1.000	2.00	0.00
Dialoogresponse	5	1.000	5.00	0.00
Aantal eenh. laden M1	6	1.000	10.00	10.00
Close files, geen logging	6	1.000	26.00	10.00
Twee recs. tijd. best.	6	1.000	2.00	0.00
Open files in M1	6	1.000	30.00	0.00
Lezen 2 recs. tijd. file	6	1.000	2.00	0.00
*				
Midden				
Zes order regels				

Fig. 44.23 Het eerste deel van het detailschema

BALIEVERKOOP

Kommentaar	Kode	Kans-faktor	Gemid. waarde	Variantie
Opzoeken art.nr.	11	1.000	60.00	10.00
Intoetsen art.nr.	1	1.000	120.00	0.00
Dialoogresponse	5	1.000	6.00	0.00
Aantal eenheden	6	1.000	54.00	0.00
Visuele kontrole	12	1.000	12.00	1.00
Intoetsen aantal	1	1.000	12.00	1.00
Dialoogresponse	5	1.000	6.00	0.00
Intoetsen 'ACC'	1	1.000	6.00	0.00
Dialoogresponse	5	1.000	6.00	0.00
Aantal eenheden	6	1.000	54.00	1.00
*				
Staart				
*				
Intoetsen 'ACC ORDER'	1	1.000	2.00	0.00
Dialoogresponse	5	1.000	1.00	0.00
Aantal eenh. (1 p.regel)	6	1.000	3.00	0.00
4 x enter + 'ACC'	1	1.000	6.00	0.00
Dialoogresponse	5	1.000	1.00	0.00
Aantal eenheden	6	1.000	1.00	0.00
UPD. rec. cycl.file	6	1.000	5.00	0.00
Close files + logging	6	1.000	26.00	0.00
Laden MO	6	1.000	10.00	10.00

Fig. 44.24 Het tweede deel van het detailschema.

BALIEVERKOOP

Parameter-overzicht

Kode rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:			
1 Invoerlengte	186.00	aanslagen	8.00
2 Transportlengte invoer	0.00	tekens	0.00
3 Transportlengte uitvoer	0.00	tekens	0.00
4 Uitvoerlengte	0.00	tekens	0.00
5 Dialoogresponses	31.00	responses	0.00
6 Dialoogsresponse-eenheden	339.00	eenheden	33.00
7 Afsluitresponses	0.00	responses	0.00
8 Afsluitresponse-eenheden	0.00	eenheden	0.00
Personele-parameters:			
11 Aanloop en uitloop	67.50	seconden	14.60
12 Wachttijden en denktijden	12.00	seconden	1.00
13 Intiksnelheid	1.50	tek./sec.	1.00
14 Foutherstelpercentage	5.00	procent	2.40
15 Min. invoer repetitietijd	0.00	seconden	0.00
16 Effektieve werktijd	70.00	procent	100.00
Machine-parameters:			
21 Transportvertraging	0.01	seconden	0.00
22 Tijd per response-eenheid	0.05	seconden	0.01
23 Afdruksnelheid	999.00	tek./sec.	0.00

Fig. 44.25 Het parameteroverzicht.

BALIEVERKOOP

Rubriek	Tussen-waarden		Aantal seconden	Variantie
Terminal Transaktie tijd (T.T.T.)				
Invoerlengte	186.00			
Intiksnelheid	1.50			
	--------	/		
Invoertijd			124.00	6837.33
Wachttijden en denktijden			12.00	1.00
Responses	31.00			
Transportvertraging	0.01			
	--------	*		
Transporttijd			0.31	0.00
Response-eenheden	339.00			
Tijd per eenheid	0.05			
	--------	*		
Verwerkingstijd			16.95	1149.29
Uitvoerlengte	0.00			
Afdruksnelheid	999.00			
	--------	/		
Uitvoertijd			0.00	0.00
Subtotaal			153.26	7987.62
Foutherstelpercentage/tijd	5.00		7.66	25.61
Aanloop en uitloop			67.50	14.60
Netto T.T.T.			228.42	8027.82
(Volledig effektief)	Standaardafwijking			89.60
Bruto T.T.T.			326.32	18556.42
(Bij 70% effektief)	Standaardafwijking			136.22

Procentuele verdeling van de T.T.T.

Invoertijd	54.29 %
Wachttijden en denktijden	5.25 %
Dialoogresponse-tijden	7.56 %
Afsluitresponse-tijden	0.00 %
Uitvoer-tijd	0.00 %
Tijd voor fout herstellen	4.76 %
Tijd voor aanloop en uitloop	29.55 %

Fig. 44.26 De pagina: Terminaltransaktietijd.

BALIEVERKOOP

Rubriek	Gemiddelde waarde		Variantie
	Lijn- en responsetijd aspekten		
Invoerrepetitietijd ongunstig	7.37 seconden		8.35
Invoerrepetitietijd normaal	10.53 seconden		19.31
Gem.berichtlengte invoer	0.00 tekens		0.00
Gem.berichtlengte uitvoer	0.00 tekens		0.00
Totaal per response	0.00 tekens		0.00
Gem. tijd per dialoogresponse	0.56 seconden		1.20
Gem. tijd per afsluitresponse	0.00 seconden		0.00
Gem. tijd per response	0.56 seconden		1.20
Gemiddelde verwerkingstijd			
per dialoogresponse	0.55 seconden		1.20
per afsluitresponse	0.00 seconden		0.00
per response	0.55 seconden		1.20
Responsetijd korter dan in X%	99%	95%	90%
Dialoogresponse	3.10	2.35	1.96
Afsluitresponse	0.00	0.00	0.00
Gemiddelde response	3.10	2.35	1.96

Fig. 44.27 De pagina: Lijn- en responsetijdaspekten.

DEEL 5
voor
transaktie-analisten

Voorkomen is beter dan genezen.
Zo langzamerhand zou iedereen
in dit vak moeten weten wat
voorkomen moet worden.

Hoofdstuk 51
Mensen, methoden, middelen

51.1 Taakomschrijving en vakmanschap

De transaktie-analist is in de eerste plaats iemand die de kursus Transaktie analyse (17) heeft gevolgd. Hij beschikt over het rekenprogramma en beheert dat. In het algemeen is de funktie transaktie-analist geen volledige baan: een transaktie-analist heeft iets anders tot hoofdtaak. Dat zou informatie-analyse kunnen zijn, maar ook systeemontwerp of netwerkbeheer. In een M3-omgeving zal het een technisch ontwerper zijn in een M1-omgeving zal een informatie-analist Transaktie analyse erbij doen.
Wie Transaktie analyse uitvoert hangt natuurlijk ook af van de gewenste resultaten. Als het alleen gaat om ergonomische resultaten, zou de informatie-analist kunnen optreden als transaktie-analist. Als het gaat om het doorrekenen van een netwerkontwerp, is een uitgebreide kennis van datakommunikatie nodig. In een M1-omgeving zou men die kennis toch al moeten inhuren voor een netwerkontwerp.
Omdat we in dit deel de logische en de technische Transaktie analyse behandelen, gaan we uit van een technisch ontwerper met voldoende kennis van datakommunikatie. Hij kommuniceert met informatie-analisten die hem de transaktieschema's leveren met de benodigde cijfers. Over die cijfers praat hij zonodig met gebruikers via de informatie-analisten.

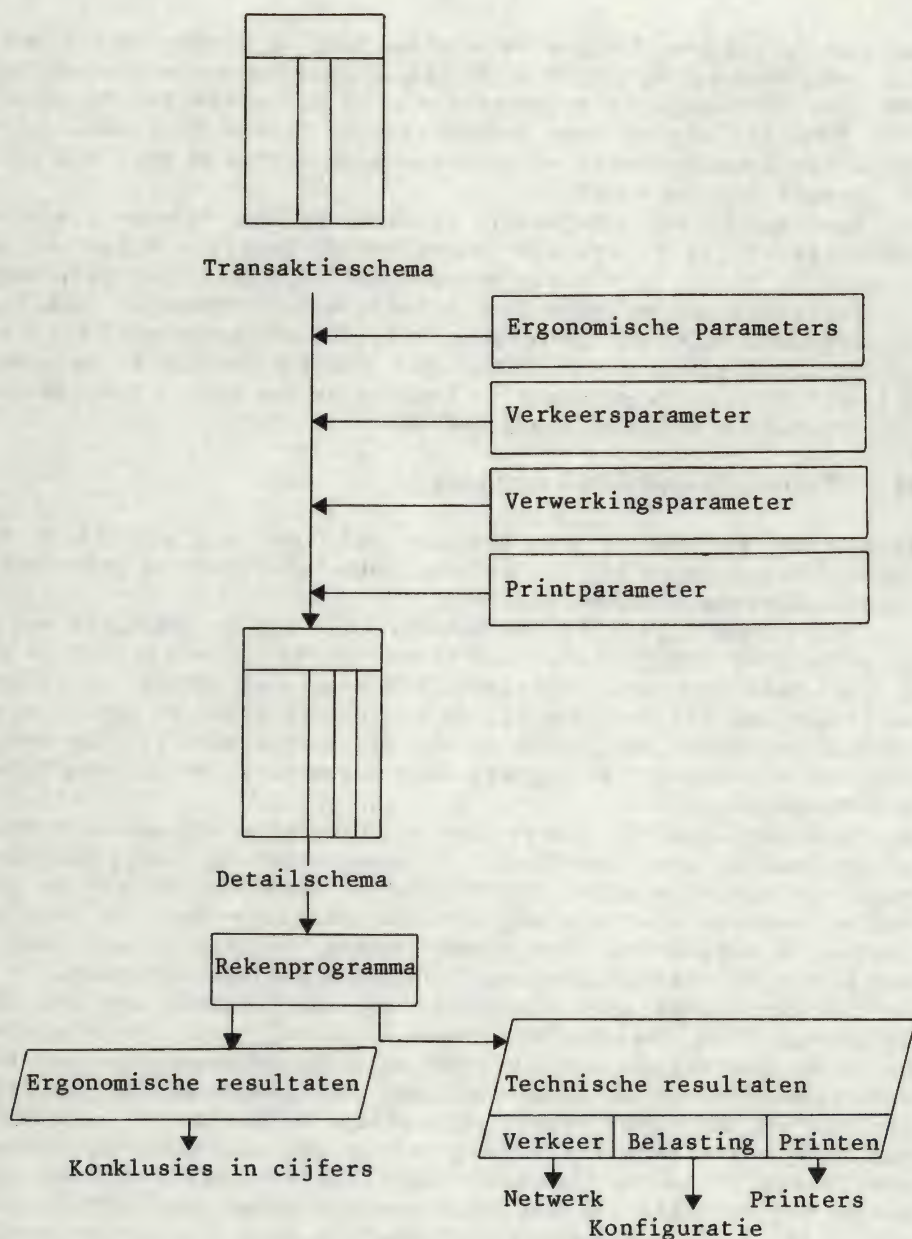


Fig. 51.1 Transaktie analyse

Om een technische Transaktie analyse uit te kunnen voeren moet hij voldoende op de hoogte zijn van de werking van access-methoden voor bestanden en databases en moet hij weten hoe de terminals bestuurd worden door de applikatie. Kortom, hij moet alles weten van datamanagement en screenmanagement van de systemen die in gebruik zijn of komen.
Als deze specialistische kennis verdeeld is over diverse systeemspecialisten, is de transaktie-analist de centrale figuur in de afstemming tussen de diverse vakgebieden. Hij houdt het responsetijdenplaatje in de gaten en speelt de konsekwenties van een ontwerp door naar de gebruikers via de informatie-analisten en draagt zonodig argumenten aan om het logisch ontwerp te herzien. Hij schrijft het rapport met de konklusies waartoe de resultaten van Transaktie analyse hebben geleid.

51.2 Transaktie analyse als methode

Transaktie analyse is een methode om eisen van gebruikers te kwantificeren en om te rekenen naar konsekwenties voor gebruiker, computersysteem en netwerkontwerp.
Als we er vanuit gaan dat de kommunikatie met de gebruiker wordt uitgevoerd door informatie-analisten dan is duidelijk dat in de methode twee soorten automatiseerders aangeduid worden: de informatie-analist die de eisen bij de gebruikers haalt en hun de konsekwenties ervan laat zien en de transaktie-analist die zorgt voor het omrekenen: van cijfers naar parameters en van resultaten naar konklusies.
De informatie-analist levert aan de transaktie-analist de transaktieschema's, een hoeveelheid cijfermateriaal en indien beschikbaar, de schermlay-outs. Of tijdens de kommunikatie met de gebruiker wel of niet dialoogsimulatie is uitgevoerd, is voor de transaktie-analist niet van direkt belang. Hij kan op basis van het geleverde transaktieschema Transaktie analyse uitvoeren.
De transaktie-analist vertaalt het transaktieschema naar een detailschema. Dat detailschema bevat cijfers, kodes, kansfaktoren en is daarmee het gekwantificeerde transaktieschema geworden. Het detailschema vormt de input voor het rekenprogramma. De resultaten die het programma levert zijn altijd hetzelfde: er verschijnen altijd drie pagina's output, maar of de resultaten van belang zijn hangt af van het gewenste doel. De transaktie-analist gebruikt bij de vertaling van het transaktieschema naar het detailschema een aantal parameters. Afhankelijk van het gewenste resultaat zal hij bepaalde parameters kiezen en andere juist weglaten. In het volgende hoofdstuk wordt dit verder uitgewerkt.
De resultaten van Transaktie analyse zijn onder te verdelen in ergonomische en technische resultaten. De ergonomische resultaten

zijn bestemd voor de informatie-analist. Wat hij daarmee doet is behandeld in het deel voor de informatie-analist. De technische resultaten vallen uiteen in belastings-, verkeers- en printcijfers en cijfers betreffende responsetijden. De belastingcijfers zijn van belang in projekten waarbij het gaat om nieuwe toepassingen op nieuwe computersystemen. Als er al erg veel toepassingen gerealiseerd zijn, is het de vraag of de belastingscijfers van direkt belang zijn. Maar ook als er niet direkt mee gewerkt wordt kunnen ze opgeslagen worden als attributen van de transakties.
De verkeerscijfers zijn van belang in alle netwerk-omgevingen.
De printcijfers zijn van belang wanneer het gaat om de synchronisatie van transakties en printwerk, het aantal printers, de bepaling van de printsnelheid en het verkeer ten gevolge van het printen in netwerk-omgevingen.
De verkeerscijfers per transaktie vormen de basis voor alles wat met het netwerk te maken heeft. Daarbij kan het gaan om een lijn, om een komplex eigen netwerk of om een openbaar netwerk. Daarmee is de witte vlek tussen logisch ontwerp en netwerkontwerp ingevuld.
De ergonomische resultaten leiden tot sociale konsekwenties in cijfers. Daarmee is de witte vlek tussen gebruikers en automatiseerders ingevuld voorzover dat nog niet door dialoogsimulatie is gebeurd.

Hoofdstuk 52
Transaktie analyse

52.1 Transaktieschema's

Transaktieschema's vormen de interface tussen informatie-analisten en transaktie-analisten. In principe zal de transaktie-analist pas detailschema's gaan maken als alle transaktieschema's binnen zijn. De kommunikatie over de kwantititeiten moet goed geregeld zijn. De informatie-analist weet welke kwantiteiten de transaktie-analist nodig heeft: dat is beschreven in het deel voor de informatie-analist. Kwantiteiten zijn aangegeven op het transaktieschema of op een begeleidend dokument. In het algemeen gesproken, is het voor de gebruiker verwarrend als er ineens iemand anders dan de informatie-analist vragen komt stellen over cijfers. Dus als er nog aanvullende cijfers nodig zijn, zal de informatie-analist kontakt opnemen met de gebruiker.
Bij de overdracht van de transaktieschema's dient de transaktie-analist vast te stellen of hij zich komplete transakties kan voorstellen. De informatie-analist zou een transaktie kunnen hebben gesplitst in subtransakties. Het is ter beoordeling aan de transakte-analist of hij een groot detailschema maakt of subdetailschema's waarvan hij de resultaten achteraf samenvoegt tot een detailschema. Zie de paragraaf Sub-transakties en kombitransakties (53.2). Als bekend is dat het gaat om decentrale of distributieve verwerking, moet worden vastgesteld in hoeverre dat

is aangegeven op de transaktieschema's. De transaktieschema's zullen wat dat betreft natuurlijk niet maatgevend zijn: transaktieschema's ontstaan tijdens het logisch ontwerp, het netwerkkoncept tijdens het technisch ontwerp. Als een ergonomische Transaktie analyse gevraagd wordt kan de transaktie-analist direkt aan het werk. Gaat het om een logische analyse dan kan het zijn dat er gewacht wordt met het maken van detailschema's tot het logisch database-ontwerp voldoende is uitgekristalliseerd. Als er een technische analyse moet worden uitgevoerd zal dat pas tijdens het technische ontwerp kunnen. In de resultaten van de logische en technische analyse is de informatie-analist geinteresseerd voorzover er konklusies uit voortkomen die de gebruiker betreffen. Die signalen kunnen de start vormen voor een iteratie in het ontwerpproces. De genoemde soorten Transaktie analyse worden behandeld in de paragraaf Vormen van Transaktie analyse (52.4).

52.2 Detailschema's

Het detailschema vormt de kwantificering van het transaktieschema en het is de input voor het rekenprogramma. Het input-deel van het standaard rekenprogramma (17) leest detailschema-regels. Wanneer de laatste regel is verwerkt, is het totaal per parameter

Detailschema

Transaktienaam: INVOER-WIJZIGING

Omschrijving	Kode	Kans-faktor	E(x)	VAR(x)
Aanloop	11	1	10	5
Intypen key	1	1	12	0
ENTER	5	1	1	0
Response-eenheden	6	1	10	0
Denk- en wachttijd	12	1	5	5
Intypen wijziging	1	1	20	20
ENTER	5	1	1	0
Response-eenheden	6	1	2	0
Uitloop	11	1	10	0

Fig. 52.1 Voorbeeld van gebruiksonvriendelijk detailschema

bepaald en begint het berekenen van de resultaten.
De linker kolom van het detailschema geeft een toelichting op de cijfers, die goed aansluit bij het transaktieschema, maar ook technischer mag zijn als het gaat om de verwerking. Het is van belang die kolom goed in te vullen: in de eerste plaats om de relatie met het transaktieschema duidelijk te houden, in de tweede plaats uit het oogpunt van dokumentatie. Het detailschema bevat soms cijfers die op zich weer het resultaat zijn van enig rekenwerk. Het is erg vervelend later niet meer te kunnen bepalen hoe een getal is ontstaan. Fig. 52.1 geeft een voorbeeld van een slecht detailschema, Fig. 52.2 geeft een betere versie.
Zoals blijkt uit de verbeterde versie mag de verwerking best worden opgesplitst in verscheidene delen. Voor het rekenprogramma maakt dat niet uit, omdat alle waarden per parameterkode worden opgeteld. In de volgende kolom staat de parameterkode: aan de hand van deze kodes wordt het rekenwerk door het programma uitgevoerd. In de derde kolom wordt de kansfaktor opgegeven. Bepaalde delen van een transaktie vinden alleen plaats in een aantal gevallen. Via de kansfaktor worden de gemiddelde waarde en de variantie gewogen opgenomen in het totaal per parameter. Wanneer in het transaktieschema termen staan als "af en toe", "soms", "in

Detailschema

Transaktienaam: INVOER-WIJZIGING

Omschrijving	Kode	Kans-faktor	E(x)	VAR(x)
Het aannemen en kontroleren van de kursuskaart en bepalen van het zoekargument.	11	1	10	5
Intypen zoekargument	1	1	12	0
ENTER	5	1	1	0
Raadplegen bestanden				
- NAW bestand	6	1	2	0
- Kursuplanning	6	1	2	0
- Lokalenplanning	6	1	3	0
- Deelnemersbestand	6	1	3	0
Intypen van de gewenste wijzigingen	1	1	20	20
ENTER	5	1	1	0
Bijwerken deelnemersbestand	6	1	2	0
Wijziging aanbrengen op kaart	11	1	10	0

Fig. 52.2 De betere versie

enkele gevallen" dan worden deze aanduidingen hier vertaald naar cijfers: de kansfaktoren. Soms wordt Transaktie analyse uitgevoerd als de dialoogstruktuurdiagrammen al bestaan. Dan kunnen die diagrammen gebruikt worden om cijfers op te noteren bijvoorbeeld uitgedrukt in kansfaktoren per scherm of per tak in de struktuur. In de vierde en vijfde kolom staan de gemiddelde waarde en de variantie van de parameter. Eenheden worden niet vermeld. Als de transaktie-analist enkele detailschema's heeft ingevuld, weet hij van iedere parameter de eenheid uit z'n hoofd. De gemiddelde waarde en variantie zijn de twee getallen waar het uiteindelijk allemaal om gaat. Zowel de gemiddelde waarde als de variantie zijn gegevens van de gebruikers, de applikatie of de technische omgeving. Met de variantie wordt aangegeven wat de afwijking van het gemiddelde is. De variantie kan echter ook gebruikt worden om de onzekerheid van de gegevens aan te geven. Als de informatie-analist vermoedt dat van een bepaald gegeven de spreiding aanzienlijk is, terwijl de gebruiker alleen een gemiddelde kan noemen, kan de transaktie-analist in zijn detailschema een stevige variantie aangeven. Die grote variantie werkt door in de resultaten en de konklusies. Als de gebruiker niet tevreden is over de nauwkeurigheid van de konklusies, dan kan hem heel eenvoudig worden aangetoond wat die onnauwkeurigheid veroorzaakt. Een getal zegt vaak meer dan tien rapporten. Wanneer de konklusies vastgelegd zijn in cijfers is de gebruiker meestal heel snel overtuigd van de noodzaak om een paar zaken eens nauwkeuriger uit te zoeken. De parameters 13-23 komen maar een keer voor op het detailschema. Het is verstandig ieder detailschema te beginnen met deze parameters.
In de paragrafen over parameters zullen de technische parameters stuk voor stuk worden behandeld. Omdat het gebruik van diverse parameters afhangt van het doel dat bereikt moet worden en van de omgeving, zullen eerst de verschillende vormen van Transaktie analyse, omgevingen en resultaten worden behandeld.

52.3 Resultaten van het rekenprogramma

Het detailschema vormt de input voor het rekenprogramma. Regel voor regel wordt door het programma gelezen. De eerste gelezen regel wordt beschouwd als de naam van de transaktie en aan de kop van iedere pagina output afgedrukt.
De eerste output van het rekenprogramma is het ingevoerde detailschema. De berekende resultaten zijn gebaseerd op deze cijfers, en een vergelijking tussen input en resultaten is dus mogelijk. Het programma kan de waarde van parameters niet kontroleren en een typefout is snel gemaakt.

ART-GEG

Lijn- en responsetijd aspekten

Rubriek	Gemiddelde waarde	Variantie
Invoerrepetitietijd ongunstig	25.61 seconden	21.71
Invoerrepetitietijd normaal	36.58 seconden	71.61
Gem. berichtlengte heen	22.10 tekens	15.96
Gem. berichtlengte terug	1142.86 tekens	75090.88
Totaal per response	1164.96 tekens	75106.81
Gem. tijd per dialoogresponse	1.38 seconden	0.00
Gem. tijd per afsluitresponse	0.00 seconden	0.00
Gem. tijd per response	1.38 seconden	0.00
Gemiddelde verwerkingstijd		
Per dialoogresponse	0.38 seconden	0.00
Per afsluitresponse	0.00 seconden	0.00
Per response	0.38 seconden	0.00

Fig. 52.3 De pagina lijn- en responsetijdaspekten.

Van iedere regel geeft de parameterkode aan bij welke tellers in het rekenprogramma de gemiddelde waarde en de variantie moet worden opgeteld, eventueel na omwerking tot 100%, als de kansfaktor kleiner is dan 1.
De eerste twee pagina's, Parameteroverzicht en Terminaltransaktietijd zijn behandeld in het deel voor de informatie-analist. De technische konklusies worden afgeleid uit de derde pagina, Lijnen responsetijdaspekten (Fig. 52.6) en voor een deel uit getallen van de eerste twee pagina's.
De T.T.T. en het aantal dialoogresponses van het parameteroverzicht maken het bijvoorbeeld mogelijk om het aantal interakties per uur te berekenen. We zullen nu eerst de derde pagina output van het rekenprogramma bespreken.
Fig. 52.3 is een voorbeeld van de pagina lijn- en responsetijdaspekten.
- Verkeersresultaten.
Het eerste gegeven dat berekend wordt is de invoerrepetitietijd: de tijd tussen twee ENTER'S. Iedere druk op de ENTER-toets betekent een start voor transport van gegevens naar de computer, voor de verwerking door de computer en voor het terugsturen van gegevens naar het beeldscherm. De tijd tussen twee ENTER'S is dus een belangrijke maat voor de belasting van zowel het computersysteem als van het netwerk.
Er worden twee waarden afgedrukt: de normale en de ongunstige. De faktor die er tussen zit is de effektiviteitsfaktor, parameterkode 16. Bij de normale invoerrepetitietijd zijn dus pauzes en persoonlijke verzorging doorberekend in de transakties. Bij de ongunstige invoerrepetitietijd gaat het om 100% effektief werken zoals dat in een pieksituatie voorkomt. Omdat de invoerrepetitietijd gebruikt wordt om belasting van netwerk en computersysteem te bepalen en omdat zowel het netwerk als het computersysteem een pieksituatie moeten kunnen verwerken, wordt in de berekeningen meestal alleen de invoerrepetitietijd ongunstig gebruikt.
Het volgende resultaat op deze pagina is de gemiddelde berichtlengte heen, de gemiddelde berichtlengte terug en het totaal. Het gemiddelde slaat op de gemiddelde lengte per interaktie, per ENTER. Wanneer het aantal· te transporteren tekens en de tranportsnelheid bekend zijn, kan gemakkelijk worden berekend hoeveel tijd het transport kost. Deze transporttijd gedeeld door de invoerrepetitietijd geeft dan de belasting aan van het transportmiddel per beeldscherm. Zo kan bijvoorbeeld worden bepaald hoeveel beeldschermen hoogstens kunnen worden aangesloten op een lijn van een bepaalde snelheid. Daarbij moet eventueel overhead ten gevolge van de lijnprocedure in rekening worden gebracht. In een bestaande situatie is de lijnsnelheid bekend en de berekening eenvoudig, zoals boven is aangegeven.

In een te ontwerpen situatie en bijvoorbeeld een gegeven aantal beeldschermen, wordt eerst een willekeurige lijnsnelheid gekozen en afhankelijk van het resultaat een hogere of een lagere lijnsnelheid. Een simpel iteratief proces. In de paragraaf Voorbeelden van kunklusies wordt dit proces uitgevoerd.
Wanneer een goede lijnsnelheid is bepaald, is de totale transportvertraging bekend. Dat is het transportdeel van de reponsetijd die immers uit twee delen bestaat: de verwerkingstijd en de transporttijd.
Als er geen netwerkontwerp gevraagd wordt, mogen de parameterkodes 2 en 3 worden weggelaten op het detailschema en zullen de berichtlengtes in het resultaat nul zijn.
De parameterkodes 2 en 3 betreffen alle tekens, inclusief die voor de schermbesturing, maar niet die van de lijnprocedure. Dat geldt dus ook voor de resultaten van het standaard rekenprogramma, zoals dat in (17) ter beschikking wordt gesteld. Natuurlijk kan in een bedrijf waar het netwerk en de lijnprocedures vastliggen een funktie aan het standaardprogramma worden toegevoegd die overhead van lijnprocedures, lijnsnelheden en wachtrij-effekten in rekening brengt.

- Responsetijden

Het volgende resultaat betreft de responsetijden . Er wordt steeds onderscheid gemaakt tussen dialoogresponses en afsluitresponses. Met een response wordt een interaktie, het drukken op de ENTER-toets bedoeld. Eerst wordt de gemiddelde tijd per response afgedrukt, dat is de som van de transporttijd en de berekende verwerkingstijd. De transporttijd is een parameter met als definitie de tijd die nodig is om een bericht heen en een bericht terug te zenden. Deze parameter wordt op het detailschema aangegeven. Wanneer het detailschema wordt gemaakt is er nog geen berichtlengte bekend dus kan er geen transporttijd berekend worden. Elke willekeurige aanname is echter voor de eerste berekening goed. Als het programma namelijk een keer de berekening heeft uitgevoerd zijn de gemiddelde berichtlengte heen en terug en het totaal bekend. Via het boven omschreven iteratieve proces kan nu een lijnsnelheid worden bepaald en de werkelijk transporttijd worden berekend. Vervolgens geeft men de parameter de waarde van deze transporttijd en laat men het programma de nieuwe resultaten berekenen. Wanneer de berekende transporttijd veel afwijkt van de oorspronkelijke schatting, dan kan de invoerrepetitietijd ook veranderd zijn en moet de berekening van de lijnsnelheid gekontroleerd worden. Daarom is het in de praktijk verstandig eerst het programma een keer uit te voeren met een willekeurige transporttijd voor de bepaling van de berichtlengte, vervolgens een schatting te doen van de transporttijd voor een redelijke lijnsnelheid op basis van de berekende berichtlengte, de parameter

voor de transporttijd aan te passen, het programma weer te draaien en dan pas het rekenwerk aan lijn en lijnbezetting uit te voeren. Resulteert dat rekenwerk in de konklusie dat de redelijke lijnsnelheid toch niet redelijk was, dan wordt voor de nu bepaalde lijnsnelheid de transportvertraging bepaald, de transportparameter aangepast en het rekenprogramma weer uitgevoerd. Het lijkt ingewikkeld maar in de praktijk is een iteratie meestal voldoende. Omdat via de T.T.T. het aantal terminals is bepaald, hoeft er geen "traffic rate table" (8) te worden gemaakt voor 1 tot n terminals!
Zoals gezegd, is de berekende responsetijd de som van deze transportparameter en de berekende verwerkingstijd.
De verwerkingstijd wordt gescheiden gehouden voor dialoogresponses, parameterkode 5, en afsluitreponses, parameterkode 7. In het detailschema wordt per interaktie de verwerking opgegeven in het aantal I/O's. Voor de verwerking ten dienste van een dialoogresponse wordt een andere parameterkode gebruikt dan voor de aanduiding van de verwerking voor een afsluitresponse. Op die manier kan het programma beide soorten verwerkingen uit elkaar houden. Een I/O heet in het detailschema een response-eenheid. Op het detailschema komt ook een parameter voor die de tijd per response-eenheid aangeeft. Op basis van deze parameters berekent het programma de totale verwerkingstijd per transaktie. Die tijd wordt afgedrukt op de pagina T.T.T. Doordat ook iedere ENTER met een response-parameter op het detailschema is aangegeven kan het programma de gemiddelde verwerkingstijd per response berekenen. Bij deze tijd wordt de waarde van de transportparameter opgeteld en dat levert de gemiddelde tijd per response op. Dat noemen we meestal de gemiddelde responsetijd. Deze twee waarden kunnen worden opgenomen als technische attributen van het entiteitstype Transakties. Vervolgens wordt, eigenlijk ten overvloede, de verwerkingstijd per response nog afgedrukt: dat is de tijd per response minus de waarde van de transportparameter.
De berekening voor de responsetijd valt dus uiteen in twee wezenlijk verschillende delen. De transporttijd wordt uiteindelijk berekend door de transaktie-analist op basis van verkeersresultaten van het rekenprogramma en de netwerksituatie. Deze berekende tijd wordt weer ingevoerd in het rekenproces door de transportparameter aan te passen.
De berekening van de verwerkingstijd gebeurt op basis van parameters in het detailschema. Hoe nauwkeuriger en realistischer deze gegevens zijn, hoe betrouwbaarder de resultaten zullen zijn. In de paragraaf Verwerkingsparameters komen we hier op terug.
De cijfers van de responsetijden dienen vergeleken te worden met de eisen van de gebruikers. Zoals altijd bij gemiddelden, moet de transaktie-analist zich realiseren waarover het gaat bij de ge-

middelde responsetijd. Als het in een transaktie om twee ENTER's gaat, waarvan de eerste een responsetijd heeft van 2 seconden en de tweede een tijd van 20 seconden, dan zou het rekenprogramma een gemiddelde responsetijd van 11 seconden berekenen. Dat is een zinloos cijfer omdat een responsetijd van 11 seconden niet voorkomt. In zo'n situatie kan de transaktie-analist, omdat hij precies weet hoe het rekenprogramma werkt, gemakkelijk aan de hand van het detailschema even beide responsetijden met de hand berekenen.
Als Transaktie analyse wordt uitgevoerd tijdens het logisch ontwerp zullen de gegevens over de verwerking nog niet erg nauwkeurig kunnen zijn. Alleen als berekende responsetijden grote verschillen vertonen met vereiste responsetijden, is het van belang de oorzaak na te gaan aan de hand van het detailschema. Hierbij valt te denken aan een faktor 10 of meer. Wanneer dat het geval is, duidt dat meestal op een fout in de analyse, want zulke grote verschillen worden namelijk nooit opgevangen door het beste technisch ontwerp. Dat betekent dat de gekozen oplossing fout is, wat meestal weer wordt veroorzaakt door onvoldoende deskundigheid of gebrek aan tijd tijdens de analyse. Daar had al op moeten vallen dat de situatie zoals die toen in kaart werd gebracht, niet op die wijze was te automatiseren met een beeldscherm. Nogmaals, het gaat om faktoren 10 en hoger, die in de praktijk regelmatig voorkomen, zonder dat er tijdens het logisch ontwerp, het technisch ontwerp, de bouw en de systeemtest, iemand alarm heeft geslagen. Dat wordt wel door de gebruiker gedaan, maar dan zijn de ontwerpers alweer bezig met het volgende projekt.
Tijdens het technisch ontwerp moet het detailschema worden aangepast. Dan zijn de gegevens nauwkeuriger en liggen ook de resultaten dichter bij de realiteit. Dan moeten opnieuw de berekende responsetijden worden vergeleken met de eisen zoals die tijdens het transaktie-ontwerp zijn vastgesteld. Dit is de laatste kans om problemen tijdens de oplevering te voorkomen. Geen enkele projektleider zal het prettig vinden, maar nu kan men voor de betrokken transakties nog terug naar het logisch ontwerp. Ook al zou dat vertraging en dus ongenoegen bij de stuurgroep of de gebruiker opleveren, het gaat om de keus tussen kleine problemen nu en grote problemen later! Veel projektleiders hebben tot hun schade moeten ervaren dat, achteraf bezien, het eerste veel beter was geweest. De verleiding te kiezen voor het laatste is velen te groot.

52.4 Vormen van Transaktie analyse

Hoewel er maar een rekenprogramma is, wordt er toch onderscheid gemaakt in de vorm van Transaktie analyse. Die vorm heeft te ma-

ken met de gewenste resultaten en daardoor ook met de parameters die op het detailschema voor moeten komen.
De eerste vorm is de ergonomische Transaktie analyse. Bij deze vorm gaat het alleen om de pagina Terminaltransaktietijd van de resultaten van het rekenprogramma. Op het detailschema komen geen parameters voor die te maken hebben met de verwerking door de applikatie of met het verkeer. Er komen alleen parameters voor die te maken hebben met de procedure aan het beeldscherm en dat zijn de kodes: 1,4,5,7,11,12,13,14 en 16. Deze kodes worden uitgebreid behandeld in (17) en zijn in het deel voor de informatie-analisten besproken om de mogelijkheden van een ergonomische Transaktie analyse duidelijk te maken. De kodes 5 en 7 werden genoemd om het verschil tussen een dialoogresponse en een afsluitresponse uit te leggen. Bij de overname van de transaktieschema's kan de informatie-analist aangeven of er afsluitresponses in de transaktie voorkomen: hij moet daarover nagedacht hebben.
Voor een ergonomische Transaktie analyse zijn die kodes op zich niet van belang omdat ze geen tijd opleveren die deel uitmaakt van de terminaltransaktietijd (T.T.T.). Als de informatie-analist echter vraagt om responsetijden in het detailschema op te nemen dan is het handig om kodes 5 en 7 op te nemen en per kode een responsetijd te simuleren met behulp van de parameters response-eenheden en tijd per response-eenheid. Bij de behandeling van de parameters komen we daarop terug. Verder is het nuttig de kodes 5 en 7 op te nemen in het detailschema omdat dan later bij een logische en technische analyse alleen de parameters 6 en 8 hoeven te worden toegevoegd. Tenslotte heeft het vermelden van parameterkodes 5 en 7 zin omdat dan op de pagina lijn- en responsetijd-aspekten, snel gekontroleerd kan worden of de door het programma berekende responsetijden kloppen met de gewenste.
Ergonomische Transaktie analyse wordt uitgevoerd om in een zo vroeg mogelijk stadium een inzicht te hebben in de T.T.T. en de gevolgen daarvan voor het aantal beeldschermen, de bezetting van de beeldschermen, het aantal beeldschermuren enzovoort. Allemaal resultaten die de informatie-analist en de gebruiker interesseren. Als op dat moment het gegevensontwerp nog niet gereed is, kan de verwerking op geen enkele manier behoorlijk in kaart worden gebracht. Vandaar dat het voorlopig goed genoeg is uit te gaan van haalbaar geachte responsetijden. Zo gauw het logisch gegevensontwerp beschikbaar is kan worden overgegaan tot de logische Transaktie analyse. Logische Transaktie analyse wordt tijdens het logisch ontwerp uitgevoerd zo gauw een gegevensmodel, een logisch database of bestandsontwerp beschikbaar is. Als het niet om een netwerkontwerp gaat hoeven nu aan het detailschema van de ergonomische analyse alleen de verwerkingsaspekten op basis van het logisch gegevensontwerp te worden toegevoegd. Als er

geen ergonomische analyse is uitgevoerd wordt het hele detailschema nu gemaakt. Ontwerpen is een iteratief proces. Dat geldt ook voor interaktieve toepassingen en de responsetijden. Natuurlijk zijn responsetijden bepaald op basis van een logisch gegevensontwerp betrekkelijk. In ieder geval komen nu de extreme responsetijden er in keiharde cijfers uit. Er zijn talloze projekten gerealiseerd met extreem lange responsetijden. Die zijn ook door het logisch ontwerp door het technisch ontwerp, door de bouw en de tests heen gekomen zonder dat iemand aan de noodrem heeft getrokken. Tijdens het logisch ontwerp kunnen responsetijden niet met een nauwkeurigheid van 100% worden voorspeld. Waar het voorlopig om gaat zijn responsetijden van 10 tot 60 seconden. Die cijfers verschijnen nu op papier. Een projektleider die dan gewoon doorgaat alsof er niets aan de hand is, zoekt problemen. Als er wel een netwerk moet worden ontworpen op basis van verkeersgegevens uit Transaktie analyse en men tijdens het logisch ontwerp een indruk van het verkeer wil hebben, dan moeten in het detailschema de verkeersparameters worden opgenomen. Met deze parameters wordt aangegeven hoeveel tekens er bij iedere interaktie naar de computer gaan en hoeveel er terugkomen. Deze hoeveelheid hangt af van de schermlayout en de dialoog. De dialoog ligt vast op het transaktieschema, de schermlay-out in de dialoogsimulator en/of op papier. Daarnaast is het soort terminal nog van belang: een domme start/stop-terminal vraagt een andere besturing dan een 3270-achtig beeldscherm.
In de praktijk doen zich twee situaties voor. In het ene geval gaat het om een projekt op een bestaand of bekend computersysteem. Dan is het soort beeldscherm een gegeven. In het andere geval moet de computer nog gekozen worden en is het soort beeldscherm onbekend. Dan kan de transaktie-analist in het detailschema het beste uitgaan van een gemiddelde, een 3270-beeldscherm en eventueel daarnaast, op een andere versie van hetzelfde detailschema van een worst case: een domme start/stop-terminal. Dat is dan meestal de grootste nauwkeurigheid die tijdens het logisch ontwerp haalbaar is.
De technische Transaktie analyse wordt uitgevoerd tijdens het technisch ontwerp. Wanneer tijdens het logisch ontwerp een logische analyse is uitgevoerd houdt de technische analyse niet meer in dan het aanpassen van verwerkingsparameters aan het technisch gegevensontwerp en van de verkeersparameters aan het uiteindelijke type beeldscherm. Als Transaktie analyse voor het eerst wordt uitgevoerd dan wordt nu in een keer het detailschema met de maximale nauwkeurigheid opgesteld. Ook nu blijft gelden dat het doel waarvoor Transaktie analyse wordt uitgevoerd bekend moet zijn, om te voorkomen dat er overbodige details in het detailschema worden opgenomen.

Uit alle drie de analyses komen dezelfde pagina's resultaten. Het verschil is de nauwkeurigheid waarmee verwerkings- en verkeersaspekten zijn opgenomen in het detailschema. Bij een ergonomische analyse zijn die aspekten helemaal niet opgenomen, er is hoogstens een responsetijd aangenomen. Bij de logische analyse zijn die aspekten bepaald op basis van logische gegevens tijdens het logisch ontwerp. Bij de technische analyse zijn ze opgenomen volgens de specifikaties van het technisch ontwerp en de technische gegevens van de beeldschermen.
De resultaten van iedere vorm van Transaktie analyse vormen altijd een samenhangend geheel. De ergonomische aspekten beinvloeden ook de technische resultaten. Als er lange denk- en wachttijden in een transaktie voorkomen, is dat terug te vinden in de T.T.T. maar ook in de resultaten betreffende het verkeer.
Tenslotte nog iets over de globale Transaktie analyse . In sommige gevallen wil men bijvoorbeeld tijdens een vooronderzoek zonder dat er details bekend zijn van de dialoog of schermlay-out, gebaseerd op globale schattingen een indruk krijgen van bijvoorbeeld het aantal beeldschermen of van de topologie van het netwerk. Op basis van globale transaktieschema's wordt dan een globaal detailschema opgesteld. De cijfers zijn uiteraard globaal. De gemiddelde waarde wordt geschat en vaak wordt de variantie niet ingevuld omdat die alleen maar zeer groot zou zijn. Dat de resultaten globaal zijn, is ook bekend zonder dat er een grote variantie verschijnt.
Als het gaat om een globale ergonomische Transaktie analyse dan komen op het detailschema ergonomische parameters voor. Wil men ook een indruk hebben van verwerking en verkeer, dan kunnen alle parameters voorkomen en gaat het dus om een globale logische Transaktie analyse. De verwerking wordt globaal geschat, maar een globale terminal bestaat niet. Er kan wel worden uitgegaan van een globaal beeldschermontwerp op een bepaald soort terminal, zoals is aangegeven bij de behandeling van de logische Transaktie analyse.
Een globale technische Transaktie analyse bestaat niet. Bij een dergelijke analyse gaat het om technische gegevens zoals die alleen tijdens het technisch ontwerp bekend zijn. Tijdens het technisch ontwerp behoren geen globale gegevens meer te bestaan, hoogstens gemiddelde waarden met een bepaalde spreiding.

52.5 Omgevingen en soorten toepassingen

In het deel voor de informatie-analist zijn een aantal omgevingen vastgesteld in het kader van transaktie-ontwerp. In verband met Transaktie analyse zijn de aspekten computerverspreiding en soort projekt van belang.

- Computerverspreiding
  - C1: Een centraal rekencentrum met een of meer mainframes met lokale beeldschermen en/of printers.
  - C1N: Idem, maar nu met een netwerk of een paar lijnen voor remote terminals.
  - C2 : Verscheidene rekencentra.
  - C2N: Idem, maar met een netwerk voor de koppeling van de systemen.
  - C3 : Een minicomputer .
  - C3N: Idem, maar dan met een netwerk voor remote terminals.
  - C4 : Een aantal microcomputers .
  - C4N: Idem, maar dan gekoppeld via een netwerk.
  - C5 : Een kombinatie van een of meer mainframes en micro's of een kombinatie van mainframess, mini's en micro's.
  - C5N: Idem, maar dan onderling gekoppeld.
- Soort projekt.
  - P1 : Het opzetten van een automatiseringplan met een globaal netwerkontwerp.
  - P2 : Netwerkontwerp op basis van gegevensdistributie.
  - P3 : Nieuwe toepassingen op bestaande systemen.
  - P4 : Nieuwe toepassingen op nieuwe hardware.
  - P5 : Nieuwe toepassingen in verband met de overgang van Cx naar CxN.
  - P6 : Overgang van batchverwerking naar interaktieve toepassingen.
  - P7 : Evaluatie van bestaande systemen.

Met bovenstaande indeling is een enorm scala van situaties aangegeven waarmee een transaktie-analist te maken kan krijgen. In grote bedrijven kunnen ze alle voorkomen. Bij kleine bedrijven gaat het meestal om een van de situaties.
Bij de toepassing van Transaktie analyse moet altijd duidelijk zijn welk doel er in de konkrete situatie bereikt moet worden: welk soort projekt in welke omgeving. Dat doel moet bekend zijn op het moment dat de detailschema's worden gemaakt. Dan moet vaststaan welke parameters moeten voorkomen. In de volgende paragrafen over parameters wordt verwezen naar deze indeling van de omgevingen.

52.6 Verkeerparameters

In deze en de volgende twee paragrafen worden de parameters behandeld die van belang zijn voor de logische en technische transaktie analyse. De overige parameters zijn behandeld in het deel voor de informatie-analist. De resultaten van elke Transaktie analyse vormen altijd een samenhangend geheel, gebaseerd op alle parameters van het detailschema.

Het aantal te transporteren tekens sterk afhangt van het soort terminal. Het is daarom zo vreemd dat in een boek als (8), dat toch als een handboek geldt, een hoofdstuk "Intelligent terminals" staat, maar dat in het deel Design calculations, waar gewerkt wordt met "the characters transmitted in a message" (pag. 401), de manier om gemiddelde berichtlengte en de spreiding ervan te berekenen wordt afgedaan met: "A study of the man-machine dialogue which takes place has given the distribution of lengths ...". Het gaat in deze pagina's om een voorbeeld, maar alle, soms zeer ingewikkelde design calculations zijn gebaseerd op de traffic volumes. Martin geeft nergens aan hoe die "study of the man-machine dialogue" resulteert in de cijfers voor al zijn berekeningen. Bij Transaktie analyse ontstaan deze cijfers als gemiddelde berichtlengte heen en gemiddelde berichtlengte terug, inclusief het tijdaspekt, als resultaat van de verkeersparameters en de ergonomische parameters op het detailschema.
Er zijn drie parameters die het verkeer betreffen

- Parameter 2: Transportlengte heen. Wanneer er op de ENTER-toets wordt gedrukt, worden er tekens van het beeldscherm naar de computer getransporteerd. Het aantal tekens hangt af van het aantal ingetoetste tekens, het soort beeldscherm en de besturing door het computersysteem. Het aantal ingetoetste tekens is bekend en wordt op het detailschema aangegeven met parameterkode 1. Het moment waarop ieder teken getransporteerd wordt is niet van belang. Bij een ongebufferde terminal gaat het teken over de lijn, direkt nadat de betrokken toets is ingedrukt. Maar ook bij dit type terminals wordt het intypen afgesloten met een toets die de funktie heeft van een ENTER-toets. Wanneer die toets wordt ingedrukt en het erbij behorende teken is getransporteerd, worden alle tekens samen door het I/O systeem overgedragen aan de applikatie en begint de verwerking.

Bij een gebufferde terminal gebeurt hetzelfde, alleen worden dan alle tekens bewaard door de terminal en verzonden wanneer op de ENTER-toets wordt gedrukt. Welke tekens er dan precies verstuurd worden hangt van de intelligentie van de terminal af. Hoewel er een groot aantal soorten terminals bestaat hoeft de transaktieanalist zich alleen maar te verdiepen in de soorten waarmee hij te maken krijgt. Iedere leverancier beschikt over dokumentatie die aangeeft wat het verband is tussen enerzijds de ingetoetste en gedisplayde tekens en anderzijds de getransporteerde tekens. Wanneer deze kennis eenmaal is opgedaan gaat het invullen van detailschema's erg snel. Als de informatie-analist op zijn transaktieschema lengtes van intoetsvelden heeft aangegeven, is de vertaling naar te transporteren tekens vrij eenvoudig.

- Parameter 3: Transportlengte terug. Als afsluiting van de verwerking door het systeem wordt een bericht naar het beeldscherm

gestuurd, dat kan bestaan uit een besturingsteken om de cursor naar het volgende veld te bewegen, maar het kan ook een heel masker zijn van honderden tekens. De tekens van de schermlay-out, zoals dat door de informatie-analist samen met de gebruiker is ontworpen, vormen het masker dat is opgeslagen in een bibliotheek, de schermbibliotheek. De manier waarop het masker is opgeborgen, hangt af van het operating system, als we even bepalen dat het operatingsystem alle software omvat, behalve de applikaties. Vaak worden maskers opgeslagen in de vorm, zoals ze ook over de lijn gestuurd worden. In dat geval is het vrij eenvoudig te bepalen hoe het masker is opgeslagen door er een print-out van te maken. Meestal is gemakkelijk vast te stellen dat er allerlei besturingstekens in de tekst voorkomen die overeenkomen met de besturingstekens voor het beeldscherm. In een P3-omgeving is het dus erg eenvoudig om even een schermlay-out te maken en vervolgens een print-out te vragen. In een P4-omgeving moet de computerleverancier die informatie verstrekken ofwel in de vorm van een handboek ofwel in de vorm van een print-out op een systeem van de leverancier. In een CxN-omgeving is deze kennis onmisbaar. Ze moet aanwezig zijn bij systeemspecialisten of bij transaktie-analisten: het detailschema moet goed ingevuld worden. Als parameterkode 2 op een detailschema voorkomt, moet kode 3 ook voorkomen. Beide komen dan een keer voor per ENTER, dus per kode 5 of 7. Op basis hiervan berekent het rekenprogramma de gemiddelde berichtlengte heen en terug plus de spreiding van beide.
- Parameter 21: Transportvertraging. Deze parameter geeft de tijd aan die nodig is om het gemiddelde bericht heen en het gemiddelde bericht terug te versturen. Als de parameterkodes 2 en 3 voorkomen, dan is het verkeer van belang en dus ook de transporttijd of -vertraging. Parameter 21 moet op het detailschema voorkomen, omdat de transporttijd deel uitmaakt van de responsetijd en van T.T.T. Als de berichtlengtes nog niet vaststaan kan de transporttijd onmogelijk berekend worden. Dit probleem wordt opgelost met de iteratieve aanpak zoals die is beschreven in de paragraaf Resultaten van het rekenprogramma. De eerste keer is de transporttijd een in principe willekeurig getal, dat echter beter zo goed mogelijk geschat kan worden. Als het rekenproces een keer is uitgevoerd, maken de berichtlengtes een goede schatting of een gedetailleerde berekening mogelijk. Dat laatste geldt meestal voor P3-situaties. In P4-situaties ontstaat het hele netwerkontwerp via een iteratief proces. Elementen in dat proces zijn dan de resultaten van Transaktie analyse per transaktie.
De parameterkodes 2 en 3 leiden dus tot resultaten, waarvan de konklusies terug te vinden zijn in kode 21. Slechts de eerste keer zijn die resultaten nog niet bekend en wordt kode 21 geschat. Wanneer de informatie-analist vraagt een ergonomisch de-

tailschema op te stellen dan worden verwerking en verkeer niet in kaart gebracht. Als de informatie-analist vraagt om alle responsetijden op bijvoorbeeld 2 seconden in te stellen dan is de snelste, maar niet de meest logische manier om dat te bereiken, parameter 21 de waarde 2 geven en alle verwerkingsparameters weg te laten. Het is de snelste manier omdat dat met een parameter geregeld kan worden. Het is niet logisch, omdat in een responsetijd altijd verwerkingstijd voorkomt en die tijd nu wordt weergegeven met een transportparameter. De ergonomische resultaten zijn echter juist. Deze instelling van de responsetijd werkt natuurlijk alleen als alle responsetijden hetzelfde zijn. Wil de informatie-analist onderscheid maken tussen bijvoorbeeld dialoogresponses en afsluitresponses dan moeten de verwerkingsparameters worden gebruikt om de juiste waarden te laten verschijnen.
De verkeersparameters hoeven niet altijd voor te komen op een detailschema. In Cx-omgevingen komen helemaal geen lijnverbindingen voor. Als in een ClN-omgeving, het netwerk bestaat uit point-to-point lijnen naar stand alone-beeldschermen, kan de lijnbelasting nooit een probleem zijn, hoogstens de transportvertraging waardoor het lang duurt voordat alle tekens van een masker op het scherm staan. Als de lijnsnelheid zo hoog is dat de transportvertraging nooit een probleem kan zijn, bijvoorbeeld 9600 bps, dan zal er bij een beeldscherm nooit een probleem zijn en niemand is dan geinteresseerd in het aantal tekens dat getransporteerd moet worden.
Als het in een ClN-omgeving gaat om point-to-point lijnen naar een cluster van beeldschermen dan is het van belang de verkeersparameters mee te nemen op het detailschema. Al zou het tijdens het logisch ontwerp allemaal nog simpel lijken, het aantal toepassingen zal groeien en het aantal beeldschermen ook. Zelfs als zou vaststaan dat de lijn met die bepaalde transaktie en het gegeven aantal beeldschermen geen probleem op kan leveren, dan is het nog van belang die situatie kwantitatief in kaart te brengen om altijd te weten hoeveel reserve kapaciteit nog beschikbaar is in het net. Als de nieuwe toepassingen ook met Transaktie analyse in kaart worden gebracht, is tijdens het technisch ontwerp al bekend of het bestaande netwerk nog bruikbaar is. Daarmee hebben we altijd problemen in een P3-situatie. Als de bestaande situatie niet in kaart is gebracht wat het verkeer betreft, dan is moeilijk te bepalen of er nog transakties of beeldschermen kunnen worden toegevoegd. In een dergelijke situatie moet eerst de bestaande situatie worden onderzocht. Dat wordt behandeld in de paragraaf Evaluatie van interaktieve toepassingen. (55.2)
Bij de verkeersparameters van Transaktie analyse gaat het om verkeer ten dienste van interaktieve toepassingen. Batch-verkeer is net zo eenvoudig in kaart te brengen als recordlay-outs van be-

standen. Batch-verkeer wordt daarom behandeld in het hoofdstuk Netwerkontwerp. Toch is er een soort batch-verkeer dat nauw verweven is met interaktieve toepassingen en dat is verkeer naar een printer. Hoewel de algemene regel is: kombineer nooit batch-verkeer met interaktief verkeer als responsetijden je lief zijn, is het niet altijd te vermijden. Een remote cluster met beeldschermen en printers is nu eenmaal een veel voorkomende situatie. In de paragraaf Andere terminals dan beeldschermen (53.6) komen we hier op terug en in het hoofdstuk Netwerkontwerp speelt batch-verkeer natuurlijk ook een rol.

52.7 Verwerkingsparameters

Op het detailschema kan ook de verwerking in kaart worden gebracht. Transaktie analyse is geen programma- of database-analyse. De verwerking wordt dus niet geanalyseerd door het rekenprogramma. De geanalyseerde verwerking kan wel in het detailschema worden opgenomen en worden meegenomen in de berekening van responsetijden. Analyse van de verwerking is zelfs op bestaande systemen nog een komplex probleem. Naast de soms ingewikkelde applikaties, zijn er nog allerlei systeemaspekten als geheugengrootte, poolsizes, het aantal schijven en het soort operating system. Transaktie analyse is een begrotingsmethode en dat betekent dat op het moment van de analyse die komplexe situatie nog niet eens bestaat!
Het benzineverbruik van een auto hangt ook van een groot aantal faktoren af. De gemiddelde autobezitter kan er al een aantal opnoemen, de autotechnicus kent zoveel details van verbrandingsmotoren dat hij er nog een reeks faktoren aan toe kan voegen. Toch geeft een autofabrikant cijfers over het benzineverbruik, soms gesplitst in cijfers die gelden op de grote weg bij een bepaalde snelheid en in cijfers die gelden in de stad. Hoewel de autotechnicus zijn hoofd schudt bij zoveel onnauwkeurigheid, zijn de cijfers toch een bruikbare basis om de kosten per kilometer te bepalen en om auto's met elkaar te vergelijken.
Iets dergelijks geldt ook voor de verwerkingsaspekten van computersystemen. De specialisten noemen onmiddelijk een reeks faktoren op waardoor de verwerkingstijd wordt beinvloed, echter zonder ze te kwantificeren. Ze geven eigenlijk alleen maar aan dat het erg ingewikkeld is. Dat zijn verbrandingsmotoren ook, al realiseert misschien niemand zich dat meer, omdat een auto een gebruiksvoorwerp is geworden. Het gemiddelde benzineverbruik is een praktische oplossing voor een ingewikkelde zaak, waar heel goed mee te werken is. Bij de verwerking door computersystemen gaat het bij Transaktie analyse om schijf-I/O's. Bij alle verwerkingen is de schijf het knelpunt. Een te klein geheugen leidt indirekt

ook tot schijf-I/O's ten gevolge van akties van het memory management. Een bekend probleem is, dat I/O kommando's van een applikatie soms niet leiden tot gegevenstransport via het kanaal. Bij batch-programma's waar vaak gesorteerde bestanden worden verwerkt, is dat heel duidelijk het geval. Bij interaktieve toepassingen waar meestal random gegevens worden opgevraagd is de kans dat het gegeven al binnen is onvoorspelbaar. Er zijn allerlei metingen gedaan naar dit soort verschijnselen om te komen tot een optimale gegevensopslag. Zolang de gebruiker echter zijn invoerdokumenten in een willekeurige volgorde mag verwerken, en wie zou dat willen verbieden, is niet voorspelbaar of er fysieke I/O's gepleegd worden of niet. Een logische I/O betreft een logische databasecall. Als het logisch database-ontwerp nog niet gereed is, beschouwen we het gegevensmodel als de logische database. Als op het transaktieschema staat dat de orderrregels van een order moeten worden gedisplayed dan gaan we ervan uit dat voor iedere regel een logische I/O nodig is als de orderregel een genormaliseerd entiteitstype is. In een non-database-omgeving beschouwen we een bestands-access als een logische I/O. In het voorbeeld van de orderregel dus een logische I/O per orderregel.
Een fysieke I/O betreft een blok op een track of een page. Het aantal fysieke I/O's wordt bepaald door de fysieke database-struktuur. Tijdens het logisch ontwerp wanneer de database-struktuur nog niet bekend is gaan we uit van logische I/O's. Het aantal I/O's kan echter sterk verminderen als de fysieke database-struktuur bekend is. Sommige systemen groeperen gegevens zodanig dat ze in een keer kunnen worden gelezen. Daarnaast kan het aantal werkelijke uitgevoerde I/O's van het operating systeem per fysieke I/O nog verschillen. Kortom, de verwerking is een volkomen ondoorzichtig en zeer komplex gebeuren. Toch leidt een verdubbeling van het aantal fysieke I/O's vaak ook tot minstens een verdubbeling van de verwerkingstijd. De responsetijd hangt af van het ontwerp qua aantal I/O's en van de tijd die voor een I/O nodig is. In de praktijk blijkt steeds weer dat ontwerpers zich geen zorgen maken over het aantal I/O's dat ze per interaktie vragen door hun ontwerp en bij lange responsetijden wijzen ze naar de tijd die het systeem per I/O nodig heeft. In een bestaande situatie kan men een meetprogramma maken dat bijvoorbeeld per kwartier meet wat een gemiddelde READ, WRITE, OBTAIN, GET of FIND kost. Daarnaast kan men meten wat het verband is tussen het aantal van die kommando's en de daarvoor benodigde tijd. Die metingen worden overall uitgevoerd: in de cijfers zitten alle komplexe systeemparameters. Zo zou men een algemene, gemiddelde verwerkingseenheid vast kunnen stellen, desnoods met een gemiddelde waarde en een spreiding en die gebruiken voor parameter 22.
In een P3-situatie is dat moeilijker, want daar kunnen geen me-

tingen gedaan worden. Dan wordt het dus schatten. In de praktijk is dat niet zo moeilijk omdat, onafhankelijk van het soort computer een gemiddelde I/O nooit nul seconden duurt en nooit 10 minuten. De grenzen liggen eigenlijk vast. Computerleveranciers doen daar meestal ook wel uitspraken over. Een goede gemiddelde waarde is 0,1 seconde voor zowel een fysieke als voor een logische I/O. Soms leiden vele logische I/O's tot een fysieke I/O en soms zijn er verscheidene fysieke I/O's nodig voor een logische I/O. Met name bij databases kan het verschil tussen logische en fysieke I/O's groot zijn, maar niet zonder dat een database-ontwerper het weet. De genoemde 0,1 seconde is dus een discutabel getal, maar in een situatie waarin niets bekend is, is het een goede start.

In de praktijk blijkt een aanname altijd te werken. Als namelijk uit een rapport blijkt dat er problemen te verwachten zijn, begint er altijd wel iemand over het aangenomen cijfer. Iedereen die dat doet mag een ander cijfer noemen. Leveranciers moeten dat kunnen, zeker als ze problemen hebben met de resultaten van het rapport.

Tenslotte is het nog zo dat de berekening van de resultaten door het rekenprogramma van Transaktie analyse een snelle gevoeligheidsanalyse mogelijk maakt. Het kost weinig moeite om van de transakties even een worst case door te rekenen. Nogmaals, elke benadering is beter dan geen benadering. James Martin zegt terecht in (8) dat 80% van de problemen met interaktieve toepassingen en netwerken voorkomen had kunnen worden als er enig rekenwerk was uitgevoerd. De praktijk leert dat het niet gebeurt omdat de gemiddelde automatiseerder teveel details weet om te kunnen denken zoals autoverkopers, die bruikbare gemiddelde cijfers produceren. Teveel gedetailleerde kennis blijkt soms een grotere handicap dan oppervlakkige kennis. Zouden performance-metingen van computerleveranciers zin hebben als er geen enkel verband bestaat tussen wat er in een applikatie aan I/O-kommando's is opgenomen en de responsetijden? De performance-grafieken van computerleveranciers bewijzen dat er een verband bestaat tussen applikatie, konfiguratie en aantal interakties per tijdseenheid. Natuurlijk moeten die grafieken met de nodige voorzichtigheid worden gebruikt, evengoed als een gemiddelde benzineverbruik nooit kan dienen om een autoverkoper met allerlei incidentele afwijkingen te konfronteren. Maar als het gemiddelde cijfer voldoende afwijkt van het in de dokumentatie vermelde, zal elke garage bereid zijn de motor te kontroleren op aspekten die het benzineverbruik beinvloeden. Het feit dat het benzineverbruik afhangt van onder andere de afstelling van de ontsteking, kan geen reden zijn niet met het gemiddelde benzineverbruik te rekenen van een goed afgestelde motor. Iets dergelijks is aan de hand, als het gaat om

performance-aspekten van computersystemen. Overbelaste slecht gedimensioneerde systemen mogen natuurlijk geen reden zijn om nieuwe systemen niet behoorlijk te begroten.
Op het detailschema wordt de verwerking in kaart gebracht met een drietal parameters:
- Parameter 6: Aantal dialoogresponse-eenheden. Een dialoogresponse is een interaktie met de computer waarvan de responsetijd kritisch is voor de gebruiker, wat veroorzaakt kan worden door zijn verwachtingspatroon. Bij bladeren verwacht iedere gebruiker responsetijden die korter zijn dan een seconde. Het kan ook veroorzaakt worden door het ritme tijdens het intoetsen. Bij massaal data entry-werk, wordt vaak blind getypt en zijn lange responsetijden al gauw onakseptabel.
Een dialoogresponse-eenheid is een zelf te kiezen eenheid van verwerking. De transaktie-analist kan, eventueel in samenwerking met de gegevensontwerper of de database-ontwerper kiezen uit een diskaccess, een bestandsaccess of een database call. Per detailschema moet alle verwerking worden uitgedrukt in deze gekozen eenheid. Dat betekent dat per parameterkode 5 (ENTER), ook parameterkode 6 voorkomt.
- Parameter 8: Aantal afsluitresponse-eenheden. Een afsluitresponse is een ENTER aan het einde van een groep aktiviteiten. Het is het moment waarop de gebruiker geneigd is een slok koffie te nemen, een vraag van een kollega te beantwoorden of iets dergelijks. De transaktie-analist stelt in overleg met de informatieanalist vast wanneer een pijl naar rechts op het transaktieschema een dialoogresponse is en wanneer het een afsluitresponse is. Het rekenprogramma houdt de verwerking ten dienste van afsluitresponses gescheiden van die welke ten dienste staan van dialoogresponses. Zoals in de paragraaf Resultaten van het rekenprogramma is beschreven worden responsetijden voor afsluitresponses berekend naast responsetijden voor dialoogresponses. De keus tussen kode 5 en 7 wordt gedaan op basis van het inzicht dat informatieanalist en gebruiker hebben in de procedure aan het beeldscherm. Bij kode 5 wordt de verwerking aangegeven met kode 6, bij kode 7 met kode 8.
Zowel parameter 6 en 8 mogen, bijvoorbeeld terwille van de duidelijkheid, gesplitst worden in verschillende regels, zoals in Fig. 52.2. Als bijvoorbeeld in een C3-omgeving bekend is dat bij bepaalde interakties eerst het programma geladen moet worden, dan kan het geen kwaad dat op een aparte regel aan te geven met als kommentaar: het laden van het programma. Met kode 6 of 8 wordt dan begroot hoeveel response-eenheden daarvoor nodig zijn.
- Parameter 22: Tijd per response-eenheid. Dit is de tijd die nodig is om een I/O uit te voeren. Het gaat om de I/O's waarin de parameters 6 en 8 zijn uitgedrukt. Per ENTER wordt dus de verwer-

king aangegeven met parameter 6 of 8, voor het hele detailschema geldt kode 22 voor de tijd per gekozen I/O. Dat maakt het gemakkelijk een aantal situaties door te rekenen voor verschillende waarden van parameter 22. Deze waarden zijn bij de behandeling van de parameters 6 en 8 besproken.

52.8 Printparameters

Er zijn twee parameters die te maken hebben met de uitvoer van de getransporteerde tekens:
- Parameters 3: Uitvoerlengte. Hiermee wordt aangegeven hoeveel tekens de computer naar de terminal stuurt. Met parameterkode 4 wordt aangegeven hoeveel tekens er getoond worden. Dat hoeft niet gelijk te zijn aan het aantal getransporteerde tekens. Enerzijds worden besturingstekens niet getoond, anderzijds kunnen door een intelligente terminals gekomprimeerde gegevens weer worden uitgebreid.
- Parameter 23: Afdruksnelheid. Met deze parameter wordt de snelheid van het afdrukken aangegeven in tekens per seconde.

Het rekenprogramma berekent de uitvoertijd, die een onderdeel is van de terminaltransaktietijd. Bij beeldschermen is de "afdruksnelheid" zo groot dat de tijd voor het tonen van het beeld vanuit de terminalbuffer te verwaarlozen is. Bij beeldschermen worden de parameter 3 en 23 dan ook niet gebruikt. Bij keyboardprinters is het gebruik uiteraard wel zinvol. Bij kombinaties van beeldscherm en printer kunnen de printparameters worden opgenomen in het detailschema van de beeldschermtransaktie als het printen een onderdeel is van de transaktie. Als een beeldscherm bijvoorbeeld gebruikt wordt in een situatie waarin per transaktie een bon geprint wordt op een printer die bij beeldscherm hoort, dan maakt de printtijd deel uit van de terminaltransaktietijd en kan het printen worden gekwantificeerd met de printparameters. In een situatie waarin verscheidene beeldschermen gebruik maken van een gemeenschappelijke printer moet per transaktie Transaktie analyse worden uitgevoerd en daarnaast moet de benodigde printtijd worden vastgesteld per transaktie op basis van het aantal printregels en de printsnelheid. Vervolgens kan gemiddeld over een dag of enkele uren worden vastgesteld hoeveel transakties op de beeldschermen zijn uitgevoerd en hoeveel printtijd daarvoor nodig is. Op die manier is bij een bepaalde printsnelheid gemakkelijk te bepalen hoe snel de printers achter gaan lopen bij de transakties. Als het printen wordt opgenomen in het detaislchema en er verkeer berekend moet worden, dan wordt met de verkeersparameter 3 ook het verkeer aangegeven. Als beeldscherm en printer via een lijn met de computer zijn verbonden moet op lijnprocedureniveau bekeken worden hoe berichten naar het beeldscherm verzonden worden en hoe

naar de printer. Vaak is de manier van versturen verschillend en is het niet juist het verkeer naar de printer met dezelfde kode 3 op te nemen als het verkeer naar het beeldscherm. Daardoor kan de gemiddelde berichtlengte terug onjuist worden. Daarom moet het printverkeer apart in kaart gebracht worden zoals alle batch-verkeer en gekombineerd worden met het interaktieve verkeer. Het is heel verstandig een keer goed uit te zoeken hoe het printverkeer verloopt ten opzichte van het interaktieve verkeer. Als die verhouding niet vanuit de applikatie of via parameters in het operatingsystem is te beinvloeden, kunnen vervelende situaties ontstaan voor de responsetijden van de interaktieve toepassingen. Als een printfile in z'n geheel over de lijn gaat kunnen er gedurende die tijd meestal geen tekens verstuurd worden voor de interaktieve toepassingen. Als een printfile wordt verstuurd in blokken die de interaktieve berichten afwisselen dan wordt het voor een groep beeldschermen aan dezelfde lijn als de printer ook al heel gauw vervelend. Soms lijkt het alsof een hardcopy-printer gekoppeld is aan een beeldscherm, terwijl in werkelijkheid de printfile over de lijn wordt geleverd door het computersysteem! Het is niet verstandig dit soort problemen maar af te wachten. Hoe het precies werkt moet uiterlijk tijdens het technisch ontwerp worden vastgesteld. Dat onderzoek begint bij de outputkommando's in de applikatie, loopt via de accesmethode en eindigt bij de lijnprocedure waar het transport uiteindelijk geregeld wordt. Belangrijke vragen zijn:

- Kan het printverkeer per regel bestuurd worden?
- Kan er een prioriteit worden aangegeven zodat printwerk als "background" getransporteerd wordt?

Waar minicomputers worden gebruikt voor clustersimulatie kan men op het mainframe "printers" konfigureren met verschillende buffergrootte. Door de mini wordt dan afhankelijk van bijvoorbeeld het aantal aktieve terminals, die "printer" aktief gemaakt die het minst storend is voor de interaktieve toepassingen. Een printer met een kleine buffer krijgt per zending minder tekens toegestuurd dan een printer met een grote buffer. Omdat het meestal gaat om een gesimuleerde printer en de regels gewoon naar de schijf gaan is de "printer" zeer snel en zeer storend. Het is ook mogelijk op de mini een vertraging op te nemen die afhankelijk is van de drukte op de lijn.
De resultaten van Transaktie analyse ten aanzien van het verkeer en de cijfers van het verkeer voor het printen maken het in ieder geval mogelijk uit te rekenen welke problemen kunnen optreden in de responsetijd en wat een aparte lijn voor het printverkeer betekent.

52.9 Benodigde tijd

Het is uiteraard onmogelijk een schatting te geven van de tijd die nodig is voor overleg met gebruikers over cijfers. Bij de onderstaande schattingen gaan we daarom uit van beschikbaarheid van de cijfers. De transaktieschema's zijn gemaakt door de informatie-analisten en hebben een definitieve vorm bereikt na dialoogsimulatie.
We zullen nu uitgaande van transaktieschema's van gemiddeld twee pagina's, een idee geven van de benodigde tijd voor de drie vormen van Transaktie analyse.
Bij ergonomische Transaktie analyse komen de verkeersparameters 3, 4 en 21 niet voor op het detailschema. De verwerkingsparameter 22 wordt altijd 1 sec. en de parameters 6 en 8 zijn steeds gelijk aan de gewenste responsetijd uitgedrukt in seconden. Wat nog enig denkwerk vraagt betreft dus eigenlijk alleen de parameters voor invoerlengte, denk- en wachttijden en aan- en uitloop. In de praktijk blijkt meestal dat het per projekt hoogstens om enkele tientallen transakties gaat die kwantitatief van belang zijn. Meestal zijn er groepen transakties die erg veel op elkaar lijken, waardoor het maken van de detailschema's steeds sneller gaat. Een richtgetal: 15-20 detailschema's per dag, uitgaande van een normale schrijfsnelheid. Als ze direkt worden ingetypt is de typesnelheid een belangrijke faktor.
Bij logische Transaktie analyse in een Cx-omgeving komt de verwerking op basis van een gegevensmodel, een logisch database-ontwerp, of de recordbeschrijvingen er nog bij. Het werkt het handigst om eerst alle detailschema's te maken zonder de kodes 6 en 8. Dan is een gesprek van hoogstens een dag voldoende om met de ontwerper van het gegevensmodel, de database of de bestanden, de verwerking in kaart te brengen per ENTER in aantallen logische I/O's van 0,1 seconde. Het richtgetal wordt dus 15-20 detailschema's per dag plus een dag voor de verwerkingsparameters. In een CxN-omgeving komen daar de parameters 3 en 4 nog bij. Na dialoogsimulatie is de schermlay-out bekend. Als het systeem nog niet gekozen is gaan we uit van een 3270-achtig beeldscherm: per veld 3 bytes extra, waarbij de vaste tekst op het scherm ook velden zijn en aaneengesloten tekst een veld is. Als de hardware bekend is moet de schermbesturing een keer worden bestudeerd maar dat valt niet onder Transaktie analyse. Het in kaart brengen van het verkeer met behulp van de kodes 3 en 4 zal het aantal detailschema's hoogstens met 20% verminderen. Het richtgetal wordt dus 12-16 per dag plus een dag voor de verwerkingsparameters.
Bij een technische Transaktie analyse verloopt alles net zoals bij de logische analyse alleen wordt de verwerking nu besproken met de technische ontwerpers. Er kan ook enige diskussie ontstaan

rond parameter 22, maar de marges zijn bekend, zoals besproken in de voorgaande paragrafen.
In een CxN-omgeving is de hardware bekend en kan de besturing van het beeldscherm nauwkeurig worden aangegeven met de parameters 3 en 4.
Als het goed is, is de technische Transaktie analyse een vervolg op de logische analyse uitgevoerd tijdens het logisch ontwerp. In dat geval hoeven op de detailschema's alleen de parameter voor de verwerking en eventueel die voor het verkeer te worden aangepast. Richtgetal: 30-40 per dag, als de detailschema's bijvoorbeeld via een line editor kunnen worden gewijzigd, omdat ze als input voor het rekenprogramma nog ergens zijn opgeslagen. Deze richtgetallen zijn gebaseerd op de ervaring met enkele tientallen projekten waar het aantal transakties uiteenliep van enkele tot veertig. De verwerking van de resultaten van de analyses vergt weer een hoeveelheid tijd die moeilijk te voorspellen is. Als er een uitgebreid rapport wordt verwacht over de ergonomische konklusies, de systeembelasting, het verkeer over de lijnen en de te verwachten responsetijden, zijn er nog wat dagen nodig om de resultaten te verwerken. In de praktijk blijkt het vaak om een of twee aspekten te gaan en is twee tot vier dagen voldoende om een goed rapport samen te stellen. Het blijft een hachelijke zaak om dit soort cijfers te noemen, omdat er nu eenmaal zeer komplexe omgevingen bestaan.
Samenvattend kunnen we vaststellen dat het Transaktie analysewerk zelf beperkt blijft tot enkele dagen. De tijd voor de voorbereiding in samenwerking met de gebruikers en het opstellen van de konklusies is moeilijk te schatten in algemeen geldende cijfers. Wel blijkt dat het totale aantal dagen een verwaarloosbare fraktie is van het totaal aantal mandagen in een projekt.

Hoofdstuk 53
Van resultaten naar konklusies

53.1 Resultaten en konklusies

Het parameteroverzicht geeft de gewogen totalen weer per parameterkode. Het is verstandig voor de andere pagina's te bekijken, eerst even door de cijfers van het parameteroverzicht heen te lopen. In feite is het parameteroverzicht de samenvatting van het detailschema. De opsteller van het detailschema moet dus snel kunnen schatten of de cijfers ongeveer kloppen. Die kontrole is nuttig omdat een typefout gauw is gemaakt.
Als het gaat om een ergonomische Transaktie analyse, worden de resultaten overhandigd aan de informatie-analist. In het deel voor de informatie-analist is behandeld wat hij er mee kan doen. Als het gaat om een logische analyse, zijn verkeer en bewerking gebaseerd op de gegevens zoals die bekend zijn tijdens het logisch ontwerp. Een logische analyse levert de verwachte technische resultaten op. De technische resultaten vallen uiteen in printresultaten, verkeersresultaten en verwerkingsresultaten.
- Printresultaten.

De printresultaten blijven beperkt tot de bepaling van de uitvoertijd op de pagina T.T.T. De aanpak bij de kombinatie van beeldschermen en printers is aangegeven in de paragraaf Printparameters en in het hoofdstuk Netwerkontwerp komen we er op terug.
- Verkeersresultaten.

De verkeersresultaten en de konklusies die daaruit kunnen worden getrokken hebben uiteraard alles te maken met het netwerkontwerp en worden dus behandeld in het hoofdstuk Netwerkontwerp. De konklusies ten aanzien van de verwerking zullen nu worden behandeld. In de paragraaf Voorbeelden van konklusies wordt aangegeven hoe de verwerkingsaspekten in kaart kunnen worden gebracht.

- Verwerkingsresultaten.

Ten aanzien van de verwerking levert Transaktie analyse twee soorten gegevens: het aandeel van de verwerking in de responsetijden en cijfers voor de bepaling van de systeembelasting. De berekening van de responsetijden is aangegeven in de paragraaf Resultaten van het rekenprogramma. (52.3)

In Cx-omgevingen bestaat de responsetijd alleen uit de verwerkingstijd. Dat betekent dat de resultaten direkt vergeleken kunnen worden met de eisen van de gebruikers zoals die bij dialoogsimulatie zijn vastgesteld. Eerst wordt gekeken naar de berekende gemiddelde responsetijd van de dialoogresponses. Als die aan de eisen voldoet wordt gekeken naar de gemiddelde responsetijd van de afsluitresponses. Bij duidelijke afwijkingen dus wanneer ze een faktor 10 of meer langer zijn dan verwacht, wordt op het detailschema gekeken naar de oorzaak. Per kode 6 of 8 wordt nog eens gekeken of de verwerking korrekt is aangegeven. Als het nog niet is gedaan kan men het rekenprogramma het detailschema nog eens laten doorrekenen voor een andere waarde van kode 22. Wanneer de situatie onbevredigend blijft, moet het gegevensmodel nog eens aan een kritisch onderzoek worden onderworpen. Wanneer dat geen oplossing is, kan de informatie-analist worden ingeschakeld om met de gebruikers de situatie nog eens door te nemen. Als de analyse van de gebruikerswensen niet verbeterd kan worden en er geen andere vorm voor de verwerking kan worden gevonden, mag de gebruiker nu zeggen of hij ondanks de verwachte responsetijden toch de transaktie wil laten bouwen.

In CxN-omgevingen bestaat de responsetijd voor een deel uit verwerkingstijd en voor een deel uit transporttijd. In een C1N-P3/P4-omgeving gaat het om een sternetwerk met vaste of geschakelde lijnen, waarvan de transportvertraging snel is te berekenen aan de hand van de resultaten van Transaktie analyse. Tijdens het logisch ontwerp zijn dit in een P4-omgeving maar globale cijfers, in een P3-omgeving kunnen de cijfers al net zo nauwkeurig zijn als in een P4-omgeving pas tijdens het technisch ontwerp het geval is. Met andere woorden, de berekende responsetijden kunnen tijdens het logisch ontwerp al aanleiding geven tot een herziening van het ontwerp. De overige CxN-omgevingen worden behandeld in het hoofdstuk Netwerkontwerp, omdat in die gevallen tijdens het logisch ontwerp nog te weinig bekend is van het netwerk om responsetijdberekeningen mogelijk te maken. Een uitzondering op

die regel is natuurlijk wanneer er van een bestaand netwerk wordt gebruik gemaakt zoals Datanet 1 waarvan de transporttijden gegeven zijn. In die gevallen kan dus ook tijdens het logisch ontwerp al een redelijke schatting van de responsetijden worden gegeven. Er blijft echter gelden dat de verwerking gebaseerd is op logische verwerkingsgegevens.
Het tweede soort gegevens omtrent de verwerking waren cijfers voor de bepaling van de systeembelasting. Computerleveranciers die met cijfers en grafieken werken hebben een aantal kengetallen die in de belasting van interaktieve toepassingen een rol spelen. De belangrijkste daarvan zijn het aantal I/O's per tijdseenheid en het aantal interakties per tijdseenheid. Andere zaken als schermlay-out, aantal invoervelden, aantal displayvelden en denk- en wachttijden zijn na dialoogsimulatie en Transaktie analyse allemaal precies bekend.
Voor iedere transaktie is het aantal I/O's per seconde gemakkelijk te berekenen door de waarden van de parameterkodes 6 en 8, zoals die zijn vermeld op het parameteroverzicht, op te tellen en de som te delen door de T.T.T.
Het aantal interakties per seconde wordt berekend door de waarden van de parameterkodes 5 en 7 uit het parameteroverzicht op te tellen en de som te delen door de T.T.T. Vermenigvuldiging van dit resultaat met 3600 levert het aantal ENTER's per uur op. Zo kunnen per transaktie deze kengetallen worden bepaald en worden opgenomen onder technische attributen van het entiteitstype Transakties. In het hoofdstuk Transakties in het deel voor de informatie-analisten is het entiteitstype Transaktie beschreven. Deze twee technische attributen vormen een belangrijke maat voor de zwaarte van transakties. De kombinatie van deze cijfers van alle transakties en alle werkplekken geeft aan hoe zwaar het systeem belast wordt door de beeldschermen. Als er op alle werkplekken maar een transaktie wordt uitgevoerd is het eenvoudig de totale belasting te bepalen: het aantal interakties per uur maal het aantal beeldschermen. Meestal ziet de praktijk er wat ingewikkelder uit: het gaat om verschillende transakties per werkplek en niet alle werkplekken zijn daarin gelijk. Nu gaat het bij de bepaling van de systeembelasting uiteindelijk om de keuze van een goede konfiguratie. In het algemeen is het te duur een konfiguratie zodanig op te zetten dat onder alle omstandigheden de gewenste responsetijden gehaald worden. Door het toevallig samenvallen van een paar zware transakties kunnen responsetijden incidenteel langer zijn dan verwacht. Het gaat erom, dat transakties die uren per dag worden uitgevoerd, met akseptabele responsetijden werken. Dat betekent dat de verdeling van de beeldschermuren per werkplek in kaart moeten worden gebracht. In het deel voor de informatie-analist is aangegeven hoe hij een tijdbestedingsdiagram (Fig.

44.7 kan maken van de beeldschermen. Dat overzicht geeft per werkplek aan hoeveel tijd er met iedere transaktie wordt gewerkt. Uit dat overzicht is direkt af te lezen, welke transakties van belang zijn voor de systeembelasting. Transakties die incidenteel voorkomen worden buiten beschouwing gelaten, hoe zwaar ze ook zijn. Dat worden de "hikken" van het systeem. Van de andere transakties worden de interakties per uur in het overzicht van de informatie-analisten opgenomen. Op die manier ontstaat een goed inzicht in het gewicht van transakties per beeldscherm, per cluster van beeldschermen en in het totaal. In de paragraaf Voorbeelden van konklusies wordt een voorbeeld van deze aanpak uitgewerkt. Het doel is de leverancier van het systeem te konfronteren met deze cijfers. In de diskussie moet natuurlijk ook batchverwerking en printwerk in rekening worden gebracht, maar die zaken kent de gemiddelde leverancier wel. Wat nieuw is, is dat ook de interaktieve toepassingen qua systeembelasting zijn gekwantificeerd en in kaart gebracht. Nu mag de computerleverancier garanties geven over de performance.
Daarmee is de situatie tijdens het logisch ontwerp beschreven. Tijdens het technisch ontwerp levert een technische Transaktie analyse uiteraard nauwkeurige resultaten. De verwerking wordt nu op het detailschema aangegeven in aantal I/O's gebaseerd op het fysieke database-ontwerp. Nog altijd gaat het daarbij om gemiddelde aantallen database-accessen. De gemiddelde tijd voor een database-access wordt zoals besproken in de paragraaf Verwerkingsparameters, vastgelegd in de parameter tijd per response-eenheid. Het aantal fysieke I/O's, dus de diskaccessen, hangt van het bekende grote aantal faktoren af. In een omgeving waar het om bestanden gaat kan best gewerkt worden met diskaccessen. Per diskaccess is een tijd van 30 milliseconden voor parameter 22 een goede aanname. Bij databases is er meestal geen inzicht in het verband tussen een logisch access en een fysieke I/O. Toch zijn van de logische accessen de gemiddelde tijden aan duidelijke grenzen gebonden. Er zijn weinig computerleveranciers die daar geen uitspraak over kunnen doen. Het gemiddelde ligt nooit onder de 0,03 seconde en meestal niet boven de 0,05 seconde. Omdat het gaat om slechts een parameter in het detailschema kunnen die situaties gemakkelijk even doorgerekend worden. Dat betekent dat de berekende responsetijden nu een stuk konkreter worden dan tijdens het logisch ontwerp. De werkwijze verloopt zoals beschreven, alleen gaat het nu niet alleen om responsetijden die een faktor 10 te lang zijn, maar een faktor 2 kan nu aanleiding zijn om het logisch ontwerp nog eens te bezien.
In CxN-P4-omgevingen kunnen, nadat het netwerk ontwerp gereed is gekomen nu vrij nauwkeurige uitspraken gedaan worden over responsetijden. Wanneer verwerkingsaspekten en netwerkaspekten van de

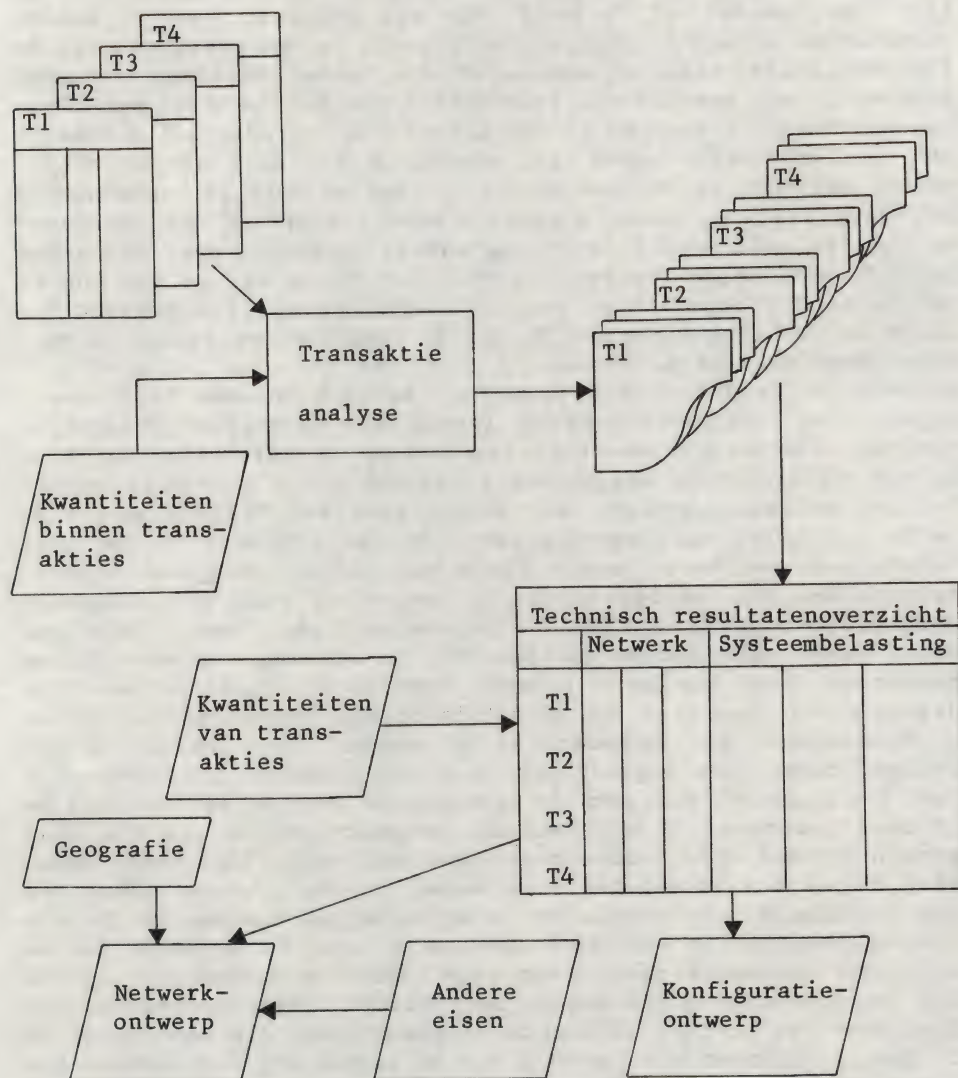


Fig. 53.1 Transaktie analyse in verband met netwerk- en konfiguratie-ontwerp.

responsetijden niet tot een akseptabele tijd leiden is er nu nog, voor het laatst, gelegenheid om terug te gaan naar de gebruiker. Er is nu in ieder geval bekend waar het knelpunt van de te lange responsetijden zit. Als het toch in de verwerking zit, moet het logisch ontwerp nog eens bezien worden. Als het in het netwerk zit dan kan dat meestal wel aangepast worden. Als de gebruiker dan meteen het prijskaartje getoond wordt mag hij nu voor het laatst beslissen. Hij akseptеert de situatie zoals die nu is ontworpen, hij betaalt voor de aanpassing aan zijn eisen voor responsetijden of hij laat de transaktie vervallen. Beter tijdens het technisch ontwerp gekeerd, dan voor altijd gedwaald! Als tijdens het logisch ontwerp overzichten zijn gemaakt van de systeembelasting, dan kunnen nu die cijfers worden aangepast aan de resultaten van de technische Transaktie analyse. Als in een P4-omgeving tijdens het logisch ontwerp een leverancier is gekozen, dan kan de konfiguratie nu nog eens besproken worden op basis van de technische belastingscijfers.
In Fig. 53.1 is het geheel in beeld gebracht.
Overzichten met cijfermateriaal kunnen uitstekend gemaakt worden met spreadsheet-programma's op microcomputers. De invloed van de wijziging van een parameter op het geheel wordt direkt zichtbaar.

53.2 Subtransakties en combitransakties

Transakties kunnen door informatie-analisten worden gesplitst in subtransakties. Dat is besproken in het deel voor de informatie-analisten. Transakties en subtransakties kunnen worden samengevoegd tot combitransakties door de transaktie-analist. Combitransakties ontstaan door de parameteroverzichten te gebruiken voor het opstellen van een detailschema.
Het samenvoegen van subtransakties tot een transaktie kan op twee manieren. De transaktie-analist kan de transaktieschema's samenvoegen tot een detailschema. Hij kan ook per subtransaktie een detailschema maken en de parameterovezichten samenvoegen tot een combitransaktie. Soms worden transakties samengevoegd tot een combitransaktie als men een gemiddelde transaktie per werkplek wil bepalen. Als op een werkplek verscheidene transakties worden uitgevoerd en de onderlinge verhouding in aantallen per dag bekend is kan men de transakties gewogen samenvoegen tot een gemiddelde transaktie. Natuurlijk worden de resultaten dan vager, maar bij een globale Transaktie analyse bijvoorbeeld, kan het nuttig zijn een totaalbeeld te schetsen. Bij een technische Transaktie analyse is het meestal beter de transakties gescheiden te houden en de resultaten te verwerken op overzichten als aangegeven in de vorige paragraaf en in Fig. 53.1. De transaktie-analist kan natuurlijk alleen combitransakties samenstellen als de gewichtsfak-

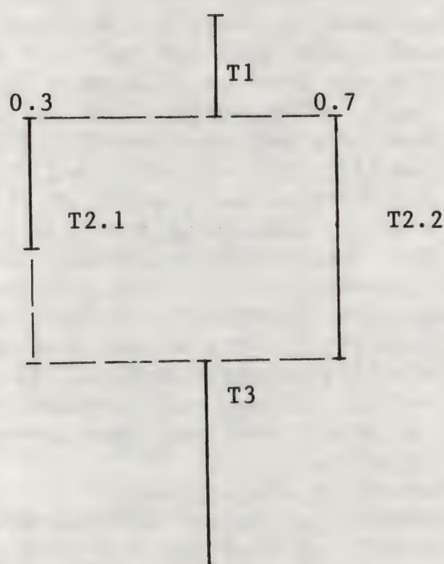


Fig. 53.2 Subtransakties.

toren van de samenstellende elementen bekend zijn. In geval van subtransakties moet de informatie-analist een struktuurplaatje leveren met daarop vermeld de percentages of de kansfaktoren. Zie Fig. 53.2. Als de transaktie-analist besluit de subtransakties op een detailschema samen te voegen dan worden T1 en T3 opgenomen met de kansfaktor 1, van T2.1 worden alle regels voorzien van de kansfaktor 0,3 en die van T2.2 van de kansfaktor 0,7.
De parameters 13-23 komen maar een keer voor, zoals op alle detailschema's, en zijn voorzien van de kansfaktor 1.
Als de transaktie-analist besluit om per subtransaktie een detailschema te maken en de resultaten samen te voegen tot een combitransaktie, dan worden de parameteroverzichten samengevoegd tot een detailschema. Van elk parameteroverzicht worden de waarden van de parameter 1 tot 12 overgenomen op het detailschema van de combitransaktie en voorzien van de juiste kansfaktor.
Deze aanpak wordt ook toegepast bij het maken van een gemiddelde transaktie. In de figuren 53.3 tot 53.5 wordt het parameteroverzicht gegeven van de samenstellende transakties. Fig. 53.6 geeft het begin weer van het detailschema van de combitransaktie. Aan het begin van dat detailschema is aangegeven hoe de kansfaktoren zijn berekend uit de frequenties.

Parameter-overzicht

Code	Rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:				
1.	Invoerlengte	17.00	aanslagen	100.00
2.	Transportlengte invoer	18.00	tekens	0.00
3.	Transportlengte uitvoer	1800.00	tekens	360120.13
4.	Uitvoerlengte	0.00	tekens	0.00
5.	Dialoogresponses	1.20	responses	0.16
6.	Dialoogresponse-eenheden	65.20	eenheden	166.56
7.	Afsluitresponses	0.00	responses	0.00
8.	Afsluitresponse-eenheden	0.00	eenheden	0.00
Personele-parameters:				
11.	Aanloop en uitloop	2.50	seconden	0.60
12.	Wachttijden en denktijden	5.00	seconden	0.00
13.	Intiksnelheid	1.00	tek./sec.	0.00
14.	Foutherstelpercentage	5.00	procent	2.40
15.	Min. invoer repetitietijd	0.00	seconden	0.00
16.	Effektieve werktijd	70.00	procent	100.00
Machine-parameters:				
21.	Transportvertraging	1.00	seconden	0.00
22.	Tijd per response-eenheid	0.10	seconden	0.00
23.	Afdruksnelheid	0.00	tek./sec.	0.00

Fig. 53.3 Het parameteroverzicht van T1.

Parameter-overzicht

Code	Rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:				
1.	Invoerlengte	115.50	aanslagen	5837.25
2.	Transportlengte invoer	124.50	tekens	6074.25
3.	Transportlengte uitvoer	750.00	tekens	362500.00
4.	Uitvoerlengte	0.00	tekens	0.00
5.	Dialoogresponses	2.50	responses	0.25
6.	Dialoogresponse-eenheden	50.00	eenheden	556.50
7.	Afsluitresponses	0.50	responses	0.25
8.	Afsluitresponse-eenheden	1.00	eenheden	0.50
Personele-parameters:				
11.	Aanloop en uitloop	5.50	seconden	4.60
12.	Wachttijden en denktijden	0.00	seconden	0.00
13.	Intiksnelheid	3.00	tek./sec.	1.00
14.	Foutherstelpercentage	5.00	procent	2.40
15.	Min. invoer repetitietijd	0.00	seconden	0.00
16.	Effektieve werktijd	70.00	procent	100.00
Machine-parameters:				
21.	Transportvertraging	1.00	seconden	0.00
22.	Tijd per response-eenheid	0.10	seconden	0.00
23.	Afdruksnelheid	0.00	tek./sec.	0.00

Fig. 53.4 Het parameteroverzicht van T2.

Parameter-overzicht

Code	Rubriek	Gemiddelde waarde		Variantie
Applikatie-parameters:				
1.	Invoerlengte	40.00	aanslagen	0.00
2.	Transportlengte invoer	52.00	tekens	0.00
3.	Transportlengte uitvoer	2475.00	tekens	286974.94
4.	Uitvoerlengte	225.00	tekens	286874.94
5.	Dialoogresponses	2.00	responses	0.00
6.	Dialoogresponse-eenheden	92.00	eenheden	96.00
7.	Afsluitresponses	0.00	responses	0.00
8.	Afsluitresponse-eenheden	0.00	eenheden	0.00
Personele-parameters:				
11.	Aanloop en uitloop	2.50	seconden	0.60
12.	Wachttijden en denktijden	0.00	seconden	0.00
13.	Intiksnelheid	1.00	tek./sec.	0.00
14.	Foutherstelpercentage	5.00	procent	2.40
15.	Min. invoer repetitietijd	0.00	seconden	0.00
16.	Effektieve werktijd	70.00	procent	100.00
Machine-parameters:				
21.	Transportvertraging	1.00	seconden	0.00
22.	Tijd per response-eenheid	0.10	seconden	0.00
23.	Afdruksnelheid	180.00	tek./sec.	0.00

Fig. 53.5 Het parameteroverzicht van T3.

Transaktienaam: T1-T2-T3-combi

Omschrijving	Kode	Kansfaktor	E(x)	VAR(x)
Transaktie Aantal kansfaktor				
T1 700 0.184				
T2 3000 0.790				
T3 100 0.026				
3800				
Voor alle transakties geldt:				
- Intiksnelheid	13	1	2	1
- Foutherstelpercentage	14	1	5	0
- Effektieve werktijd	16	1	70	100
- Transportvertraging	21	1	1	0
- Tijd per response-eenheid	22	1	0.1	0
- Afdruksnelheid	23	1	180	50
T1----------------------------				
Invoerlengte	1	0.184	17.00	100.00
Transportlengte invoer	2	0.184	18.00	0.00
Transportlengte uitvoer	3	0.184	18000.00	360120.13
Dialoogresponses	5	0.184	1.20	0.16
Dialoogresponse-eenheden	6	0.184	65.20	166.65
Afsluitresponses	7	0.184	0.00	0.00
Afsluitresponse-eenheden	8	0.184	0.00	0.00
Aan- en uitloop	11	0.184	2.50	0.60
Denk- en wachttijden	12	0.184	5.00	0.00
T2----------------------------				
Invoerlengte	1	0.79	115.50	5837.25
Transportlengte invoer	2	0.79	124.50	6074.25
Transportlengte uitvoer	3	0.79	750.00	62500.00
Dialoogresponses	5	0.79	2.50	0.25
Dialoogresponse-eenheden	6	0.79	50.00	556.50
Afsluitresponses	7	0.79	0.50	0.25
Afsluitresponse-eenheden	8	0.79	1.00	0.50
Aan- en uitloop	11	0.79	5.50	4.60
Denk- en wachttijden	12	0.79	0.00	0.00
T3----------------------------				
Invoerlengte	1	0.026	40.00	0.00
Transportlengte invoer	2	0.026	52.00	0.00
Transportlengte uitvoer	3	0.026	2475.00	286974.94
Dialoogresponses	5	0.026	2.00	0.00
Dialoogresponse-eenheden	6	0.026	92.00	96.00
Afsluitresponses	7	0.026	0.00	0.00
Afsluitresponse-eenheden	8	0.026	0.00	0.00
Aan- en uitloop	11	0.026	2.50	0.60
Denk- en wachttijden	12	0.026	0.00	0.00

Fig. 53.6 Detailschema van de combitransaktie

53.3 Relatie met andere projektaktiviteiten

Transaktie analyse kan altijd worden ingepast in de reeds toegepaste methoden, omdat het altijd een aanvulling is. Wel moet ervoor gezorgd worden dat de afstemming met de andere aktiviteiten goed is geregeld.
Een belangrijk synchronisatiepunt is het gereedkomen van alle transaktieschema's. Een projektleider hoort dat meetpunt te kennen. Hij moet in ieder geval bij de overdracht van de transaktieschema's aanwezig zijn. Eigenlijk moet hij deze gebruikersdokumenten nauwkeurig bestuderen. Het geheel van alle transaktieschema's vormt de gebruikersvisie op het hele interaktieve gebeuren van het projekt. Die gebruikers komen straks afrekenen. Tijdens de overdracht stelt de transaktie-analist vast of de transaktieschema's voldoende gegevens bevatten over kwantiteiten. Zo niet, dan moet op dat moment worden vastgesteld, hoe die gegevens boven water komen. Het kan zijn dat de informatie-analist alleen maar hoeft te verwijzen naar de data dictionary/directory waar de transakties als entiteiten zijn opgenomen en alle gegevens over gegevens zijn opgeslagen. Detailschema's worden projektdokumenten. Vaak zullen detailschema's worden ingevoerd in een computersysteem als input voor het rekenprogramma. Vanaf dat moment is het detailschema als dokument overbodig geworden. De regelmatige aanpassing van het schema in de verschillende vormen van Transaktie analyse vragen natuurlijk ook om opslag in een computersysteem. De toegang tot de detailschema's zou voor het wijzigen voorbehouden moeten blijven aan de verantwoordelijke transaktie-analist. Wat geldt voor de detailschema's, geldt ook voor de pagina's output van het rekenprogramma. De belangrijkste resultaten worden als attributen opgenomen bij de transakties.
Als Transaktie analyse deel uitmaakt van transaktie-ontwerp is de synchronisatie met andere projektaktiviteiten zoals dat behandeld is in het deel voor de informatie-analist. Als Transaktie analyse moet worden uitgevoerd om een aantal transakties te kwantificeren, hangt de werkwijze af van de fase waarin het projekt verkeert en het doel van de analyse. Het te bereiken doel bepaalt tevens de vorm van Transaktie analyse.
- Tijdens het logisch ontwerp. Als het gaat om een ergonomische analyse worden verwerking en verkeer niet in kaart gebracht en is de aanwezigheid van transaktieschema's het enige kriterium. Als het gaat om een logische Transaktie analyse omdat men een indruk wil hebben van de te verwachten resultaten, moet er een gegevensmodel of logisch database-ontwerp beschikbaar zijn om de logische verwerking te kunnen schatten. De verwerking moet op het transaktieschema zijn aangegeven en moet worden gekontroleerd aan de hand van het gegevensmodel. In een CxN-omgeving kan desgewenst

ook een aanname gedaan worden van de transportvertraging door het netwerk.
- Tijdens het technisch ontwerp moet het fysieke database-ontwerp bekend zijn, omdat van een technische Transaktie analyse verwacht wordt dat die de nauwkeurigste technische resultaten oplevert. Op basis van die resultaten wordt besloten tot GO of NO GO voor de bouw.
In een P1-omgeving zijn er nog niet zoveel relaties met projekt-aktiviteiten omdat de projekten nog geselekteerd moeten worden. Wanneer er reeds corporate meta-gegevens beschikbaar zijn kan daarvan bij het opzetten van de detailschema's uitstekend gebruik gemaakt worden.
In een P2-omgeving kan tijdens het logisch ontwerp niet altijd de verwerking in kaart worden gebracht omdat vaak nog niet bekend is in welke vorm de gegevens beschikbaar zullen worden gesteld via het nog te ontwerpen netwerk. Er kunnen meestal wel enkele situaties als uitgangspunt worden genomen. In ieder geval blijft de ergonomische Transaktie analyse mogelijk. Tijdens het technisch ontwerp. zal Transaktie analyse herhaaldelijk worden uitgevoerd voor alle transakties die gedistribueerde gegevens verwerken. De bepaling van de konfiguratie verloopt in dat geval eveneens als een iteratief proces. Er kan een type computer gekozen worden dat nog uitgebreid kan worden. Aan het eind van het netwerkontwerp kan met een laatste technische Transaktie analyse bepaald worden wat de systeembelasting wordt. Dan kan de konfiguratie worden aangepast. Het dialoogontwerp en de dialoogsimulatie behoren onafhankelijk te blijven van de plaats waar de gegevens zijn opgeslagen en worden dus gewoon tijdens het logisch ontwerp uitgevoerd.
In P3-omgevingen is de situatie tamelijk eenvoudig. Nu kan tijdens het logisch ontwerp al met technische gegevens worden gewerkt omdat het type beeldschermen, de databasehandler en het operating system bekend zijn. Wanneer ze niet bekend zijn in de vorm die nodig is voor Transaktie analyse dan kan nu de benodigde kennis worden verzameld zoals is aangegeven in de paragrafen over parameters. De logische Transaktie analyse is nu in sommige opzichten het zo nauwkeurig als de technische Transaktie analyse.
In P4-omgevingen bestaat altijd een spanningsveld tussen de beschikbaarheid van de specifikaties en het kiezen van een computerleverancier. De aanpak zou als volgt moeten verlopen. Tijdens het logisch ontwerp wordt, als onderdeel van het transaktie-ontwerp, de logische Transaktie analyse uitgevoerd. Op basis van het geografische plaatje wordt de topologie van het netwerk getekend. Meestal kan het netwerk dan nog op een aantal manieren worden opgebouwd. Onderzoek welke datakommunikatie-funkties daarvoor nodig zijn en zoek de leveranciers die dat kunnen. De leveranciers die

garanties willen geven op basis van de belastingcijfers doen mee in de selectieprocedure, die verder op de bekende manier verloopt. Dus tijdens het logisch ontwerp een kwalitatieve keus voor een bepaald type computer. Tijdens het technisch ontwerp kan de konfiguratie worden aangepast onder andere op basis van de technische Transaktie analyse.
In een P5-omgeving worden bestaande systemen gekoppeld, waardoor nieuwe toepassingen mogelijk worden. De hardware is bekend en daarmee is de situatie gelijk aan P3-omgevingen. Het netwerkontwerp vindt plaats tijdens het technisch ontwerp op basis van de technische Transaktie analyse. De netwerkaspekten worden omgerekend naar konsekwenties voor de responsetijden zoals is aangegeven in de paragraaf Resultaten van het rekenprogramma.
De P6-situatie lijkt altijd eenvoudig maar betekent, enkele uitzonderingen daargelaten, een volledig nieuw ontwerp. Of men met de bestaande of met nieuwe hardware gaat werken maakt daarbij niet veel uit. Zowel de beeldschermen als de gegevensopslag zijn nieuw. Dat betekent dat de aanpak dezelfde is als in de P4-omgevingen, misschien dan afgezien van de keuze van de leverancier. Details van het ontwerpen van netwerken worden uiteraard behandeld in het hoofdstuk Netwerkontwerp.

53.4 Transaktie analyse in distributieve omgevingen

Op basis van de vastgestelde omgevingen is gemakkelijk aan te geven wat er verstaan wordt onder centraal, decentraal en distributief. Zowel de C1-omgevingen als de C1N-omgeving noemen we wel een centrale omgeving. Daarmee is alvast aangegeven dat het al dan niet aanwezig zijn van een netwerk niets afdoet aan centraal zijn of niet-centraal zijn. De Cx-omgevingen, waarin x gelijk is aan 2,3,4 of 5 zijn decentraal. De CxN-omgevingen zijn alleen distributief als applikaties op gekoppelde systemen afhankelijk van elkaar zijn in die zin, dat ze op ad-hoc-basis gegevens moeten uitwisselen. Dat betekent dat interaktieve toepassingen enerzijds konverseren met beeldschermen en anderzijds met andere computersystemen. Dit noemen we interaktief verkeer tussen computersystemen. Daarbij is dan nog niet vastgesteld in welke vorm de gegevens worden uitgewisseld. Dat kan een record zijn uit een bestand, het kan ook een uittreksel uit een database zijn in de vorm van een bestand. In het laatste geval spreken we van interaktief batch-verkeer. Karakteristiek voor interaktief verkeer tussen computersystemen is dat het deel uitmaakt van een responsetijd. Een gebruiker drukt op de ENTER-toets en start de verwerking door het computersysteem. Die verwerking bestaat minimaal uit lokale verwerking, maximaal uit lokale verwerking, interaktief verkeer en verwerking op een ander computersysteem. In prin-

cipe zouden de laatste twee elementen zelfs verscheidene keren kunnen voorkomen.
Wanneer er sprake is van batch-verkeer tussen computersystemen ten dienste van batch-verwerkende programma's gaat het om een decentrale omgeving.
In een zo'n omgeving wordt verkeer eenvoudig bepaald door de grootte van de records en het aantal records. Dat soort berekeningen wordt al uitgevoerd sinds we bestanden ontwerpen en daar is Transaktie analyse niet voor nodig.
Bij Transaktie analyse gaat het om interaktieve toepassingen en de belasting op de resources waaronder ook de koppeling tussen de computersystemen valt. Bij de belasting op de resources gaat het om situaties die te vaak voorkomen om incidenteel genoemd te kunnen worden. Responsetijden die in 90 tot 95% van de gevallen aan de eisen voldoen, zijn in de meeste gevallen akseptabel. De te lange responsetijden zijn incidenteel en het is meestal de moeite niet ook die aan de eisen te laten voldoen. We zullen nu het voorgaande toelichten aan de hand van twee praktijksituaties.
- In een netwerk van microcomputers wordt gewerkt met een fileserver. Deze fileserver wordt 's morgens via een vaste lijn naar een mainframe voorzien van nieuwe bestanden die de lokale verwerking op de diverse micro's mogelijk maken. Het verzenden van de bestanden wordt uitgevoerd door een programma dat via het beeldscherm van de fileserver wordt gestart. We nemen vervolgens aan dat tijdens het verzenden van de bestanden alle toepassingen die gebruik maken van bestanden van de fileserver, geblokkeerd zijn. Er moeten 1000 records van 80 bytes worden verstuurd.
- Een minicomputer is via een vaste lijn gekoppeld aan een mainframe. Op de mini worden via een tiental beeldschermen transakties uitgevoerd die gebruikmaken van de lokale gegevens. Als een bepaald gegeven niet aanwezig is in het lokale bestand wordt het opgevraagd bij het mainframe. Is het gegeven ook centraal niet beschikbaar, dan verschijnt de mededeling op het scherm van de gebruiker dat het bewuste gegeven niet voorkomt in de bestanden. Gebruikers schatten dat het in 10% van de transakties zal gaan om gegevens die niet lokaal beschikbaar zijn.
Hoewel men naar de letter in beide gevallen kan spreken van een interaktieve toepassing, die verkeer tussen systemen veroorzaakt, dat onderdeel is van de responsetijd, bestaat er een groot verschil tussen beide situaties. In het eerste geval gaat het meer om het starten van een batch-programma dan om een interaktieve toepassing. Gezien de mogelijke resultaten, is Transaktie analyse in het eerste geval dan ook niet zinvol. De belasting van de lijn en de fileserver benaderen de 100% en dat is zonder Transaktie analyse uitstekend vast te stellen. Als de grootte van het bestand bekend is, is het niet moeilijk te bepalen, wat bij een be-

paalde lijnsnelheid de transporttijd zal worden.
In het tweede geval is niet zo eenvoudig te bepalen wat de bezetting van de lijn zal worden en hoe de responsetijd zal worden. Zelfs al is bekend hoeveel tekens per opvraging bij het mainframe verzonden worden, dan moet nog bepaald worden welke lijnsnelheid vereist is en welke invloed die snelheid dan heeft op de totale responsetijd die de gebruiker ervaart.
Beide voorbeelden geven aan, hoe moeilijk het is om situaties te karakteriseren. De een kan volhouden dat het in beide gevallen om interaktieve toepassingen gaat, een ander meent dat het gaat om een batch-toepassing en een interaktieve toepassing. De transaktie-analist zal zeggen dat hij alleen in het tweede geval Transaktie analyse zinvol acht om het verkeer tussen de mini en het mainframe in kaart te brengen.
In de paragraaf Resultaten van het rekenprogramma is behandeld welke resultaten het rekenprogramma levert ten aanzien van het verkeer. De berekening van de gemiddelde berichtlengtes is gebaseerd op de parameterkodes 2 en 3. Met deze parameters wordt verkeer tussen beeldscherm en computer aangegeven per ENTER. Er zijn geen parameters om in het detailschema het verkeer tussen computersystemen aan te geven. Anders gezegd: van slechts een verbinding kan het verkeer geanalyseerd worden en dat is de verbinding tussen beeldscherm en computer. De analyse van het verkeer tussen computersystemen is een aparte aktiviteit waarbij wel gebruik gemaakt wordt van de resultaten van Transaktie analyse.
Bij de analyse van het verkeer gaat het om het aantal tekens dat per seconde getransporteerd moet worden. Transaktie analyse levert de gemiddelde berichtlengte en de tijd tussen de transporten: de invoerrepetitietijd.
Op basis van deze gegevens kan het verkeer en dus de lijnbezetting worden bepaald. Bij verkeer tussen computersystemen is de berichtlengte, zeker tijdens het technisch ontwerp, bekend. De gegevens die tussen twee computersystemen worden uitgewisseld hebben meestal niets met de dialoog van een transaktie te maken. Het zijn gewoon gegevens van of naar een (remote) database. De tijd tussen twee transporten hangt af van de terminaltransaktietijd. Als er in alle transakties eenmaal per transaktie transport over de verbinding tussen de computersystemen plaats vindt, is voor die verbinding de terminaltransaktietijd de invoerrepetitietijd. Als er slechts in 10% van de transakties kommunikatie met het mainframe plaats heeft, is de invoerrepetitietijd 10 x de terminaltransaktietijd. De rest van de berekeningen verloopt zoals is aangegeven in de paragraaf Voorbeelden van konklusies: bij een gegeven aantal beeldschermen kunnen de lijnsnelheid en lijnbelasting bepaald worden, bij een gegeven lijnsnelheid kan de transporttijd en dus de uiteindelijke responsetijd worden bere-

kend.
Tenslotte moet dan het iteratieve aspekt nog aangegeven worden. Hoe moet in het detailschema de distributieve verwerking in de verschillende stadia van het interaktieve proces worden aangegeven? De vraag kan ook als volgt worden geformuleerd: hoe wordt een decentraal transaktieschema, dus met 5 kolommen, vertaald naar het detailschema?
Voor de gebruiker bestaat er alleen een responsetijd, onafhankelijk van het soort verwerking. Het transport naar en de verwerking op een ander computersysteem moet worden uitgedrukt in parameters die de verwerking aangeven: de parameters 6 en 8. Voor de transportparameter 21 wordt in eerste instantie een willekeurige waarde aangenomen. Dat wil zeggen dat we een transporttijd aannemen en die later, als de berichtlengtes bekend zijn, aanpassen. Hetzelfde geldt voor de tijd voor het transport tussen de computersystemen. De aangenomen waarde moet echter worden uitgedrukt in de parameters 6 of 8. Hoeveel tijd deze parameters voorstellen is aangegeven met parameter 21.
Een voorbeeld: de tijd voor het transport tussen de computersystemen wordt geschat op 1 seconde. Parameter 22 heeft de waarde van 0,1 seconde. De tijd voor het transport tussen de systemen wordt dan op het detailschema aangegeven met: kode 6, kans: 1, E(x):10, VAR(x):0. Het geheel is in Fig. 53.7 in beeld gebracht.

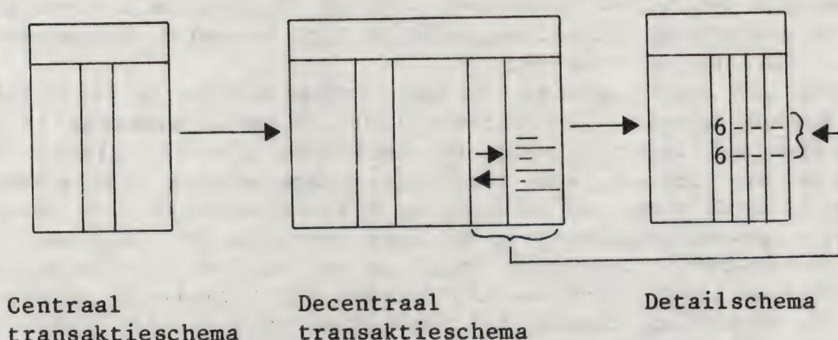


Fig. 53.7 Distributieve zaken op een detailschema.

Op de volgende regel moet natuurlijk nog de verwerking op het andere systeem worden aangegeven, eveneens met kode 6 of 8. Kortom, het hele distributieve gebeuren wordt uitgedrukt in response-eenheden van de transaktie die geanalyseerd wordt en komt zodoende korrekt tot uitdrukking in de responsetijden. Wanneer nu een lijnsnelheid wordt bepaald, gebaseerd op het verkeer, zoals aangegeven in de paragraaf Verkeersparameters, kan de werkelijke tijd voor het transport tussen de systemen worden bepaald en kan het detailschema worden aangepast. Het rekenprogramma levert dan de verbeterde resultaten.
In de beschreven situatie ging het steeds om een verbinding tussen computersystemen. In de praktijk zal het vaak gaan om een netwerk. Voor alle vormen van netwerken geldt echter dat de kapaciteit gebaseerd moet zijn op de hoeveelheid verkeer die getransporteerd moet worden. Transaktie analyse levert de gegevens die nodig zijn om die hoeveelheden te bepalen: aantal tekens per tijdseenheid.
Een ander aspekt is de mate waarin computersystemen belast worden door distributieve toepassingen. In het geval van een microcomputernetwerk met een fileserver zal het transport over de verbinding misschien niet van belang zijn, maar hoe staat het met de belasting van de fileserver? De terminaltransaktietijd in kombinatie met het aantal response-eenheden per transaktie op de fileserver levert de belasting in cijfers, zoals dat is beschreven in de paragraaf Verwerkingsparameters.
In het hoofdstuk Netwerkontwerpmethode wordt verder uitgewerkt hoe aan de hand van de resultaten van Transaktie analyse in een distributieve omgeving een netwerk wordt ontworpen.

53.5 Transaktie als entiteitstype

In de meeste bedrijven ontstaan interaktieve toepassingen doordat men kiest voor beeldschermen en vervolgens interaktieve programma's bouwt. Gewoonlijk kwantificeert men niet, maar vertaalt men alleen de dialoog naar programma's en randombestanden. Als er al gekwantificeerd wordt gaat het meestal aan "transactions on databases" (7). Gestadig wordt het aantal interaktieve toepassingen uitgebreid en groeit het aantal beeldschermen. Op het moment dat de responsetijden niet meer akseptabel zijn weet niemand wat er nu precies aan de hand is. Vaak komen computerleveranciers met voorstellen voor uitbreiding van de hardware, maar ze garanderen nooit welke verbetering de uitbreiding brengt.
Daarom is het van belang te allen tijde, ook in P3-omgevingen, de transakties zoals ze in eerste instantie op transaktieschema's zijn beschreven, als entiteiten op te nemen in de systeemdokumentatie of de datadictionary/directory.

Entiteitstype: Transaktie
- Algemene attributen
  - Transaktienaam
  - Status
  - Schermen
  - Transaktiedefinitie
  - Lokatie(s)
  - Printer (J/N)
- Ergonomische attributen
  - Frequentie
  - Pieken
  - Terminaltransaktietijd (T.T.T.), tijdens dialoogsimulatie
    - Totaal
    - Intiktijd
    - Aan- en uitlooptijd
    - Denk- en wachttijd
    - Printtijd
    - Overige
  - Terminaltransaktietijd (T.T.T.), berekend met Transaktie analyse:
    - Totaal
    - Intiktijd
    - Aan- en uitlooptijd
    - Denk- en wachttijd
    - Printtijd
    - Overige
  - Beeldschermuren per dag:
  - Aantal beeldschermen:
  - Aantal printers:
  - Lokatie van printer(s):

Gegevensgebruik
- Technische attributen
  - Response-eenheden per transaktie
  - Response-eenheden per seconde
  - ENTER's per uur
  - Gemiddelde dialoogresponsetijd
  - Gemiddelde afsluitresponsetijd
  - Printregels per uur
- Netwerkattributen
  - Gemiddelde berichtlengte heen
  - Gemiddelde berichtlengte terug
  - Invoerrepetitietijd ongunstig
  - Gemiddelde dialoogresponsetijd
  - Gemiddelde afsluitresponsetijd
  - Kommunikatie met andere systemen
    - Batch (J/N)

- Interaktief (J/N)

In het deel voor de informatie-analist zijn de algemene en ergonomische attributen besproken, we zullen nu de technische en netwerkattributen behandelen.

- Response-eenheden per transaktie. Op het parameteroverzicht is aangegeven het totaal van dialoogresponse-eenheden en afsluitresponse-eenheden. Dit attribuut is de som van beide.
- Response-eenheden per seconde. Deze waarde wordt gevonden door het vorige attribuut te delen door de T.T.T. Aangezien het bij de systeembelasting vaak gaat om de pieksituaties die het systeem nog goed moet verwerken, nemen we de netto T.T.T.
- ENTER's per uur. Dit is de som van het aantal dialoogresponses en afsluitresponses uit het parameteroverzicht gedeeld door de T.T.T., vermenigvuldigd met 3600.

Het aantal response-eenheden per seconde en het aantal ENTER's per uur zijn belangrijke gegevens voor de bepaling van de konfiguratie. Deze cijfers zijn karakteristiek voor de zwaarte van transakties. Ze komen meestal voor in de performance-grafieken van computerleveranciers. In de paragraaf Voorbeelden van konklusies wordt aangegeven hoe deze cijfers per transaktie worden omgerekend naar de totale situatie.

- Gemiddelde tijd per dialoogresponse. Dit gegeven wordt overgenomen van de pagina lijn- en responsetijdaspekten.
- De gemiddelde tijd per dialoogresponse. Ook deze waarde wordt overgenomen van de genoemde pagina.

Deze attributen kunnen na een technische Transaktie analyse een andere waarde hebben dan na een logische analyse.

- Printregels per uur. Wanneer per transaktie een hoeveelheid printwerk moet worden uitgevoerd, als onderdeel van de transaktie of parallel aan het uitvoeren van beeldschermtransakties, wordt hiermee aangegeven om hoeveel regels per uur het gaat. Details zijn reeds besproken in de paragraaf Printparameters.
- Gemiddelde berichtlengte heen
- Gemiddelde berichtlengte terug
- Invoerrepetitietijd ongunstig.

Deze drie attributen worden weer overgenomen uit de pagina Lijn- en responsetijdaspekten.

- Gemiddelde dialoogresponsetijd:
- Gemiddelde afsluitresponsetijd:

Deze responsetijden worden berekend door van de pagina Lijn- en responsetijdaspekten de gemiddelde verwerkingstijd per dialoogresponse en per afsluitresponse te nemen en daar de vertraging van het netwerk bij te tellen. Dat kan dus pas als het netwerk is bepaald. Deze transporttijd mag ook worden weergegeven met parameterkode 21 en dus worden meegenomen in Transaktie analyse. In dat geval zijn deze attributen gelijk aan de gemiddelde tijd per

dialoogresponse resp. per afsluitresponse. Het doel van deze attributen is om, onafhankelijk van Transaktie analyse het effekt van een netwerk op de responsetijden vast te leggen.
- Kommunikatie met andere systemen. Batch: J, wil zeggen dat er batch-verkeer, gestart door deze transaktie, plaatsvindt met een ander systeem. Karakteristiek voor batch-verkeer is, dat het geen deel uitmaakt van de responsetijd. Het transport verloopt niet-synchroon met de transaktie. Batch:N wil zeggen dat door deze transaktie geen batch-verkeer wordt gestart. Als er dus, door een transaktie een herhaaldelijk uit te voeren, een bestand ontstaat, dat later verzonden wordt dan hoort het batch-verkeer thuis bij de transaktie die het transport start. Als dat niet door een transaktie gebeurt maar bijvoorbeeld door het systeem zelf op basis van de klok, dan hoort dit verkeer bij het batch-verkeer dat onafhankelijk van transakties in kaart gebracht wordt bij het netwerkontwerp.
Dit attribuut is er om aan te geven dat voor deze transaktie het batch-verkeer nog in kaart gebracht moet worden. Batch-verkeer valt echter buiten Transaktie analyse. Bij netwerkontwerp komt het weer ter sprake.
Interaktief J betekent dat er per transaktie kommunikatie plaats vindt met een ander systeem, hetzij in de vorm van een bericht heen en terug, hetzij in de vorm van een bericht heen en een bestand terug. Dat betekent niet, dat het in alle gevallen bij iedere ENTER gebeurt. In het detailschema kan per betrokken ENTER een kansfaktor aangeven in hoeveel procent van de gevallen het voorkomt. Het interaktieve verkeer is gebonden aan de transaktie, en maakt meestal deel uit van de responsetijd, ook al zou het transporteren geheel of gedeeltelijk parallel lopen met bijvoorbeeld het intypen. Interaktief en batch-verkeer kunnen natuurlijk beide voorkomen. Interaktief verkeer wordt opgenomen in het detailschema, batch-verkeer wordt apart in kaart gebracht.
Wanneer de transakties op deze manier in kaart worden gebracht, kunnen met allerlei geautomatiseerde hulpmiddelen overzichten worden gemaakt. Zeker bij grote aantallen transakties en verscheidene vestigingsplaatsen is het geheel zonder gestruktureerde informatie niet meer te overzien.
In de paragraaf Voorbeelden van konklusies en in het hoofdstuk Netwerkontwerp zullen een aantal mogelijkheden worden gegeven om de attributen van transakties te gebruiken. Op de hulpmiddelen die daarbij gebruikt kunnen worden zal niet verder worden ingegaan. Zowel zij die beschikken over een data dictionary/directory als de eigenaars van de PC's met spreadsheets worden geacht de mogelijkheden te kennen van deze hulpmiddelen voor het maken van tijdbestedingsdiagrammen, systeembelastingsoverzichten en dergelijke.

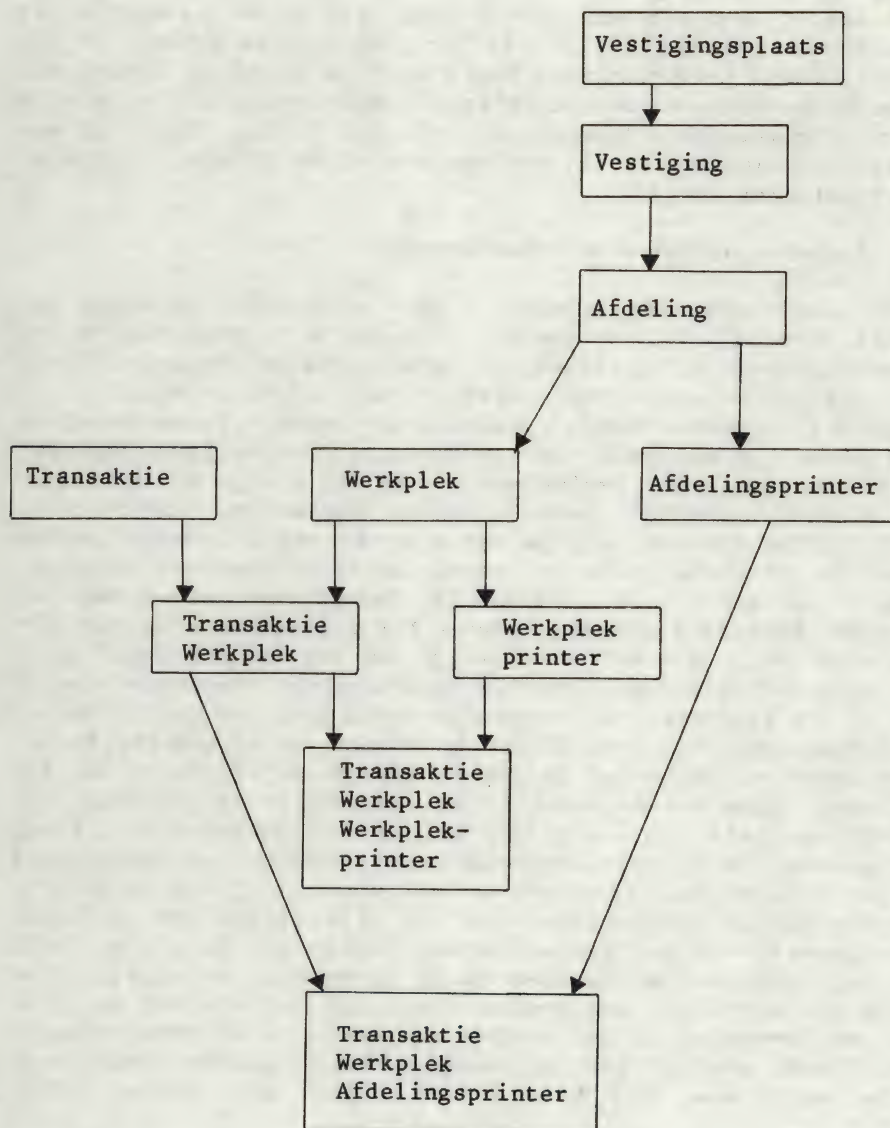


Fig. 53.8 De transaktie in een geografisch gegevensmodel.

Om netwerken te kunnen ontwerpen moeten gegevensgebruik, gegevensopslag en de geografische situatie met elkaar in verband worden gebracht. Het gegevensgebruik ligt vast in de transaktie, de gegevensopslag is een lokatie in het geografische geheel. In het hoofdstuk over netwerkontwerp komen we hier op terug. De relatie tussen transakties en de geografische situatie zou men in de vorm van een gegevensmodel kunnen weergeven als in Fig. 53.8. Dit model kan gebruikt worden bij het opslaan van de genoemde gegevens in een datadictionary.

53.6 Andere terminals dan beeldschermen

Voor interaktieve toepassingen is het beeldscherm de meest gebruikte terminal. Daarom worden in bijna alle situaties beeldschermen genoemd als intermediair tussen mens en computer. Transaktie analyse is echter ook bruikbaar voor andere apparatuur.
- Printers. We maken hierbij onderscheid tussen key-boardprinters en printers. In het geval van key-boardprinters verloopt Transaktie analyse zoals bij beeldschermen, alleen worden nu de parameters 4 en 23 gebruikt, naast de andere parameters, zie de paragraaf Printparameters. Bij printers wordt het printen opgenomen in het detailschema zoals is aangegeven in de genoemde paragraaf of het wordt apart in kaart gebracht. Daarbij kan het om twee zaken gaan: de tijd die nodig is voor het printen en voor het verkeer over de lijn. Het is mogelijk om een detailschema op te stellen voor het printen. Daarin komen dan de parameters 4 en 23 voor en als bovendien het verkeer berekend moet worden, parameter 3. Doordat er geen interakties voorkomen is de gemiddelde berichtlengte nu het totaal aantal tekens. De T.T.T. is nu de tijd die nodig is om een dokument te printen. Op basis van deze tijd kan bijvoorbeeld onderzocht worden hoeveel printers per groep beeldschermen nodig zijn, bij welk aantal transakties de printers nog juist in de pas lopen met de beeldschermen of hoe de responsetijden van de beeldschermen zullen zijn tijdens het printen.
- Leespennen. Met een leespen worden tekens gelezen, soms in de vorm van barkodes. Het bewegen van de leespen is een routine handeling die het beste kan worden uitgedrukt in seconden en in de vorm van parameter 12 kan worden opgenomen in het detailschema. Daarbij moet meestal een spreiding worden opgenomen, omdat soms de leesaktie niet goed verloopt en opnieuw moet worden uitgevoerd.

53.7 Transaktie analyse en Datanet 1

Bij lijnverbindingen wordt gelet op de hoeveelheid verkeer vanwege transportvertraging en lijnbelasting, maar niet vanwege de

kosten. Bij een netwerk is de hoeveelheid verkeer maatgevend voor de benodigde kapaciteit binnen het netwerk.
Bij gebruik van een pakketgeschakeld net als Datanet 1 blijven er slechts twee zaken over, als het om verkeer gaat: de bezetting van de aansluiting en de volumekosten. De bezetting van de aansluiting wordt berekend zoals dat gebeurt voor elke vaste lijn en wordt beschreven in het volgende hoofdstuk.
Bij de bepaling van de volumekosten gaat het om de hoeveelheid verkeer per transaktie. Transaktie analyse levert het aantal interakties per transaktie en de berichtlengtes per interaktie uitgedrukt in tekens.
Er hoeft dus alleen een vertaalslag te worden uitgevoerd van berichtlengtes naar aantal segmenten van 64 bytes, de rekeneenheid binnen Datanet 1. In (44) wordt een voorbeeld uitgewerkt dat in grote lijnen alsvolgt verloopt.
- Gegevens: de gemiddelde lengte van een bericht en de variantie, beide resultaat van Transaktie analyse.
- Bereken de kans P(n) dat het bericht past in n segmenten. Begin met n=1 en als de kans P de 100% nadert is n groot genoeg.
- Bereken het gemiddelde aantal segmenten alsvolgt: 1 x P(1) + 2 x P(2) + 3 x P(3) + ... n x P(n). Aan de hand van dit gegeven wordt vervolgens aangegeven hoe het aantal pakketten kan worden berekend om de lijnbelasting van de aansluiting te berekenen, De invoerrepetitietijd is hier uiteraard weer een belangrijk gegeven. Omdat het segment de rekeneenheid van de PTT is, kunnen nu ook de volumekosten per transaktie in kaart worden gebracht.
In het algemeen maakt men zich geen zorgen over de hoeveelheid verkeer over een lijn omdat de tarieven voor telekommunikatielijnen vaste bedragen per maand zijn, onafhankelijk van het gebruik. Bij Datanet 1 echter, moet iedere byte betaald worden. Dat betekent dat het, meer dan vroeger zin heeft zich te verdiepen om de hoeveelheid verkeer per transaktie. Daarbij gaat het om de volgende zaken:
- Logische redundantie. Stuur bijvoorbeeld niet de komplete NAW-gegevens over bij adreswijzigingen maar alleen de delen die veranderd zijn.
- Bestandsredundantie. Bij opslag op schijf kijken we niet op een paar trailing blanks. Vaak worden programma's zo gebouwd dat ze klakkeloos items uit bestanden oversturen naar beeldschermen. Een dump van de blokken die over de lijn gaan kan verhelderend werken!
- Technische redundantie. Bij de juiste besturing van beeldschermen is het vaak mogelijk de hoeveelheid getransporteerde bytes te minimaliseren. Natuurlijk speelt het type terminal daarbij een rol. Op dat aspekt komen we terug in de paragraaf 54.3: Netwerkontwerp en verkeer.

53.8 Voorbeelden van konklusies

In paragraaf 44.7 hebben we de ergonomische resultaten van Transaktie analyse vertaald naar konklusies die van belang zijn voor de gebruikers. In deze paragraaf gaat het om konklusies uit de technische resultaten. Voordat we een paar voorbeelden uitwerken, eerst iets over de zuiverheid van de konklusies. In de meeste situaties gaat het niet om een transktie op een beeldscherm. Het gaat meestal om groepen werkplekken en per werkplek een groep transakties. De resultaten van Transaktie analyse ontstaan per transaktie. Met die resultaten kan men twee kanten op: voor een beperkt aantal transakties of combinaties van transakties de konklusies exakt berekenen of een gewogen gemiddelde transaktie vaststellen en als combitransaktie door het rekenprogramma laten doorrekenen en zo tot gemiddelde konklusies komen. Het werken met combitransakties is behandeld in paragraaf 53.2. We zullen ons nu beperken tot het vaststellen van konklusies voor een beperkt aantal transakties. Er zijn verschillende invalshoeken om tot een selektie te komen.

- Als het gaat om de benodigde kapaciteit van het netwerk of de konfiguratie, worden de zwaarste transakties per werkplek geselekteerd.

Het gewicht van een transaktie is terug te vinden in de resultaten van Transaktie analyse: verkeer en verwerking

- Naast het gewicht is meestal ook het aantal uren per beeldscherme per dag een basis voor de selektie. Transaktie die maar een paar minuten per dag gebruikt worden zijn minder belangrijk dan transakties die vele uren per dag uitgevoerd worden.
- Bijna altijd geldt een 20-80 regel: 80% van het werk wordt gedaan met 20% van de transakties. Natuurlijk zal de verhouding per projekt anders zijn, maar er zijn altijd een aantal transakties die bijna niet gebruikt worden en waar niemand de kapaciteit van het netwerk of de grootte van de konfiguratie op af wil stemmen.

Hoewel het niet noodzakelijk is, is het verstandig het aantal te beschouwen transakties te beperken.

We pakken het eerste voorbeeld in paragraaf 44.7 weer op. Op basis van de ergonomische resultaten van Transaktie analyse is per afdeling het aantal beeldschermen bepaald en per beeldscherm het aantal uren per transaktie. In de tabel van Fig. 53.9 staan de transakties in de eerste kolom en de resultaten in de volgende kolommen.

- Netto T.T.T. In het resultatenoverzicht voor de informatie-analist hebben we de bruto T.T.T. gebruikt om het aantal beeldschermen en beeldschermuren te bepalen. De technische resultaten betreffen de belasting van het netwerk en de konfiguratie. Dan moet erop gerekend worden dat de terminals voor 100% effektief

Transaktie	Netto T.T.T.	Interakties (kode 5+7)	Interakties per uur	Resp.eenh (kode 6+8)	Resp.eenh per sec.
T1	562.29	123.18	720	91.68	0.16
T2	307.11	44	504	66.2	0.21
T3	29.63	6	720	6	0.2
T4	28.79	5	648	28	0.97
T5	167.76	29	612	35.1	0.21
T6	21.97	3	504	29	1.3
T7	12.11	2	576	8	0.66
T8	42	5	396	22	0.52
T9	42	5	396	22	0.52
T10	50	5	360	37	0.74
T11	55	5	324	38	0.69
T12	29.99	8	936	8	0.27
T13	151.48	23.1	540	18.9	0.12
T14	29.99	8	960	8	0.27
T15	521.34	67.9	432	8	0.01
T16	131.32	27	720	29	0.22
T17	30	2	144	25	0.83
T18	35	2	180	7	0.2
T19	598.36	68.2	396	45.89	0.07
T20	14.99	8	1800	2	0.13
T21	82.87	5.4	540	22.8	0.27
T22	115.11	3	72	16	0.14
T23	17.24	4	828	16	0.93
T24	17.15	4.6	972	21.3	1.24
T25	15.45	7	1656	20	1.29
T26	61.81	11	648	7	0.11
T27	14.32	2.4	612	22.4	1.56
T28	76.42	21	972	164	2.14
T29	132.44	46.05	1224	31.05	0.02
T30	556.33	121	782	17	0.03

Fig. 53.9 Verwerkingsresultaten

bezet kunnen zijn en dus rekenen we hier met de netto T.T.T.
-Interakties. Per transaktie kunnen we het aantal keren dat op de ENTER-toets wordt gedrukt, eenvoudig vaststellen door de waarde van parameterkode 5 en 7 uit het parameteroverzicht bij elkaar te tellen.
- Interakties per uur. Deze waarde wordt alsvolgt uit de twee voorgaande kolommen berekend: (kode 5 + kode 7) x 3600/netto T.T.T.
- Response-eenheden. De verwerking wordt bij Transaktie analyse uitgedrukt in response-eenheden. Het aantal response-eenheden van een transaktie wordt gevonden door de parameterkodes 6 en 8 uit het parameteroverzicht op te tellen.
- Response-eenheden per seconde. De waarden in de vorige kolom worden gedeeld door de netto T.T.T. en leveren zo het aantal response-eenheden per seconde. Per transaktie per beeldscherm moet het systeem dus dit aantal response-eenheden per seconde kunnen verwerken en bij n beeldschermen, n x zoveel.
Het aantal interakties per uur en het aantal response-eenheden per seconde zijn de bepalende grootheden voor de "zwaarte" van interaktieve toepassingen voor computersystemen. Bij metingen achteraf worden dan ook altijd dit soort zaken vastgesteld. Bij toepassing van Transaktie analyse zijn deze cijfers in voorlopige vorm beschikbaar tijdens het logisch ontwerp en in definitieve vorm tijdens het technisch ontwerp. Dan is er nog gelegenheid genoeg om een konfiguratie aan te passen, zie paragraaf 33.2.
Uit het tijdsbestedingsoverzicht dat de informatieanalist heeft gemaakt, zie Fig. 44.7, blijkt hoeveel uren men per dag de verschillende transakties uitvoert. We kunnen nu een aantal situaties doorrekenen om de worst case te bepalen welke de konfiguratie nog ruimschoots moet kunnen trekken. We beginnen daarom met de transakties die enige uren per dag worden uitgevoerd en waarvan we in Fig. 53.9 zien dat ze tot de zwaarste behoren. Met de stippellijnen hebben we een paar van die situaties aangegeven, zie Fig. 53.10. Voor deze situaties kunnen we nu gemakkelijk vaststellen welke belasting het systeem te verwerken krijgt in ENTER's per uur en in I/O's per seconde, zie Fig. 53.11. Bij de vaststelling van de konfiguratie krijgt de computerleverancier de beschikking over deze cijfers. Daarmee zijn de interaktieve toepassingen, die altijd het ongrijpbare deel van de systeembelasting vormen, vastgelegd in cijfers zoals computerleveranciers die zelf ook hanteren.
Batch-programma's en printwerk moeten natuurlijk ook in kaart gebracht worden, maar de invloed daarom moet de computerleverancier kunnen aangeven. Die verwerking wordt volledig door het systeem bepaald.
Ten aanzien van het verkeer geldt in grote lijnen dezelfde aan-

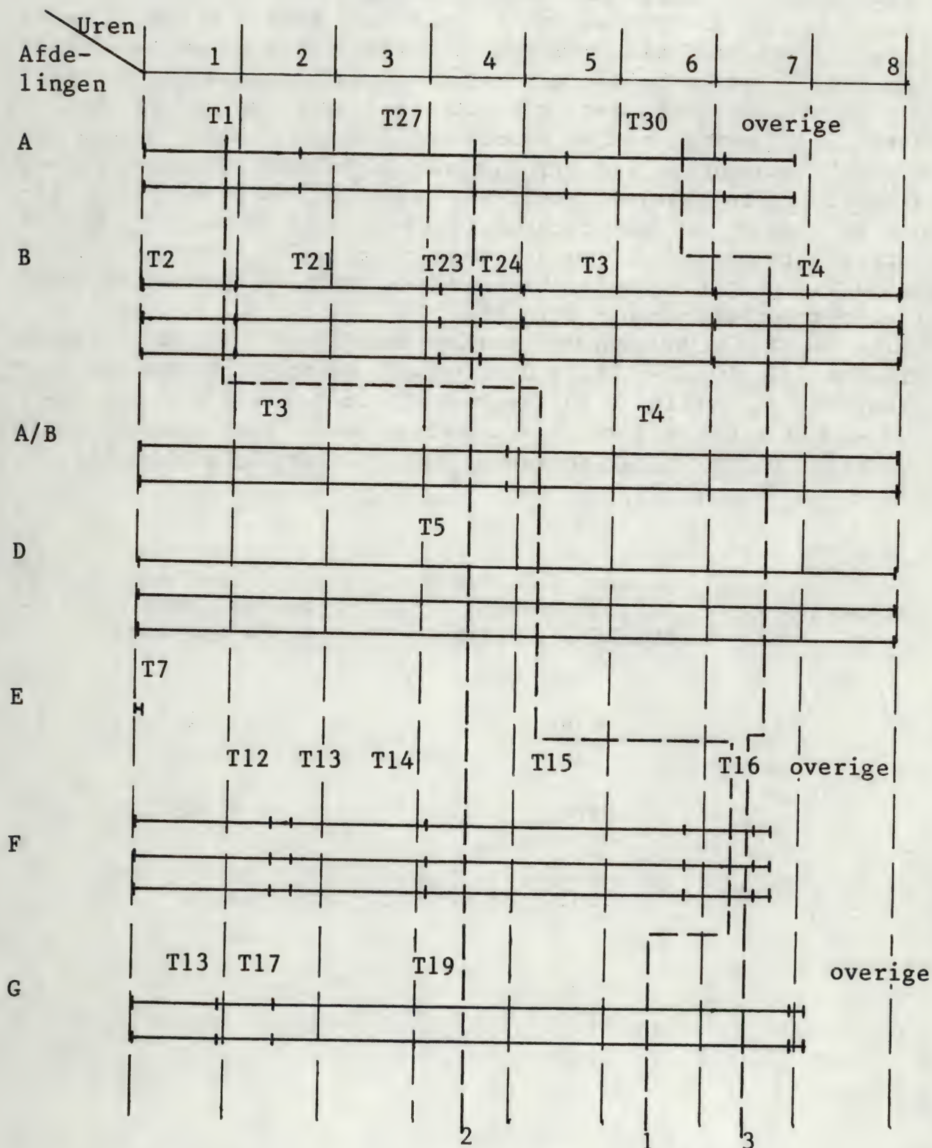


Fig. 53.10 Resultatenoverzicht.

pak. Per transaktie is het verkeer nu bekend. Er wordt een overzicht gemaakt als Fig. 53.9, maar nu met de invoerrepetitietijd, de gemiddelde berichtlengte heen en terug. Voor een gegeven lijnsnelheid kan nu per transaktie de belasting worden berekend in %. Als de lijnsnelheid niet gegeven is wordt er een aangenomen en op basis van de totale belasting, zoals die nu berekend kan worden, wordt de aanname aangepast: het bekende iteratieve proces.
Omdat nu het aantal beeldschermen per lokatie bekend is kan ook beoordeeld worden of van een gegeven lijn naar een cluster nog een ander cluster, bijvoorbeeld via een multi op aansluiting, gebruik kan maken. Er kan beoordeeld worden wat er gebeurt als er printers gebruik maken van een lijn naar beeldschermen. Men kan konstant in de gaten houden hoe een bestaande datakommunikatie-infrastruktuur zijn grens bereikt.
Details van die berekening worden behandeld in het volgende hoofdstuk. In Fig. 53.11 is het totaal weergegeven van de systeembelasting. Daarin is bijvoorbeeld 9036 = 2 x 720 + 3 x 504 + 2 x 648 + 3 x 612 + 3 x 720 + 2 x 396. De overige getallen zijn op dezelfde manier berekend aan de hand van de cijfers in Fig. 53.9.

Situatie	Totaal aantal ENTERS's per uur	Totaal aantal I/O's per sec.
1	9036	4.74
2	9072	7.11
3	9592	6.34

Fig. 53.11 Totaal generaal.

Hoofdstuk 54
De netwerkontwerpmethode

54.1 Netwerken en geografie

In de onderstaande tabel is nog eens het verband aangegeven tussen werkplekken en transakties. De geografie moet tijdens de analysefase in kaart zijn gebracht. Tijdens het logisch ontwerp zijn de transakties ontworpen. Om aan te geven over welk netwerkontwerp het in de volgende paragrafen gaat, hebben we de systemen en de soorten netwerken naast de geografische indeling gezet. Het verband tussen de lokatie en het systeem is aanvechtbaar: er zullen zeker uitzonderingen bestaan. Hetzelfde geldt voor de verdeling van een vestiging of een kantoor in afdelingen. Natuurlijk zijn er vele andere niveau's denkbaar, maar qua netwerk gaat het in ieder geval om een of meer cluster of een LAN. De tabel is echter bruikbaar om aan te geven om welke omgevingen en netwerken het gaat.
De vestigingsplaats is een plaatsnaam. Vestigingsplaatsen worden door een WAN met elkaar verbonden. Per vestigingsplaats zal het gaan om een of om verscheidene vestigingen/kantoren. Bij verscheidene vestigingen kan gebruik gemaakt worden van lokale PTT lijnen. Voor het verkeer zijn lokale lijnen gelijk aan interlokale, maar voor lokale lijnen zijn de PTT-tarieven lager en vaak kunnen goedkopere modems worden toegepast.
Binnen een vestiging zijn werkplekken te onderscheiden die veelal gegroepeerd zijn binnen de funktionele eenheden. Deze clusters

Geografie	Systemen per lokatie	Onderlinge verbindingen	Transaktie-ontwerp
Vestigings-plaatsen	Mainframe, mini	WAN	
Vestigingen, kantoren	Mainframe, mini	lokale lijnen	
Afdelingen, groepen	Mini, cluster	LAN, cluster-controller, concentrator	
Werplekken	Beeldschermen	Micronetwerk	Transakties

Fig. 54.1 Netwerken en geografie.

zijn dan meestal aangesloten op een clustercontroller die verbonden is met het netwerk. Als binnen de afdeling een minicomputer in gebruik is, zijn alle beeldschermen daarop aangesloten via coax-kabels of V24-kabels. Bij aanwezigheid van verscheidene mini's kan de veelheid aan kabels vervangen zijn door een LAN. In een kleinschalige omgeving kan het per kantoor gaan: een micronetwerk dat de micro's op de werkplekken verbindt. Evenals een clustercontroller kan ook zo'n micronetwerk zijn aangesloten op het WAN of de lokale lijnen.
Uiteraard komen niet altijd alle in de tabel genoemde situaties voor. Bij koppeling tussen microcomputers en mainframe komt het afdelingsniveau niet voor. Bij sommige bedrijven komen alleen micro's voor.
We zullen nu de netwerksituaties van de geschetste omgevingen relateren aan de resultaten van Transaktie analyse.

- Bij netwerken binnen een vestiging, een LAN of een micronetwerk, zijn de verkeersresultaten niet van belang. De systeembelasting door interaktieve toepassingen blijft van belang, onafhankelijk van de manier waarop de beeldschermen op het systeem zijn aangesloten.
- Bij lokale lijnen en een WAN zijn de verkeersresultaten van Transaktie analyse direkt bruikbaar voor het netwerkontwerp.
- In C1N/C3N-omgevingen levert Transaktie analyse cijfers voor het netwerkontwerp en de systeembelasting.

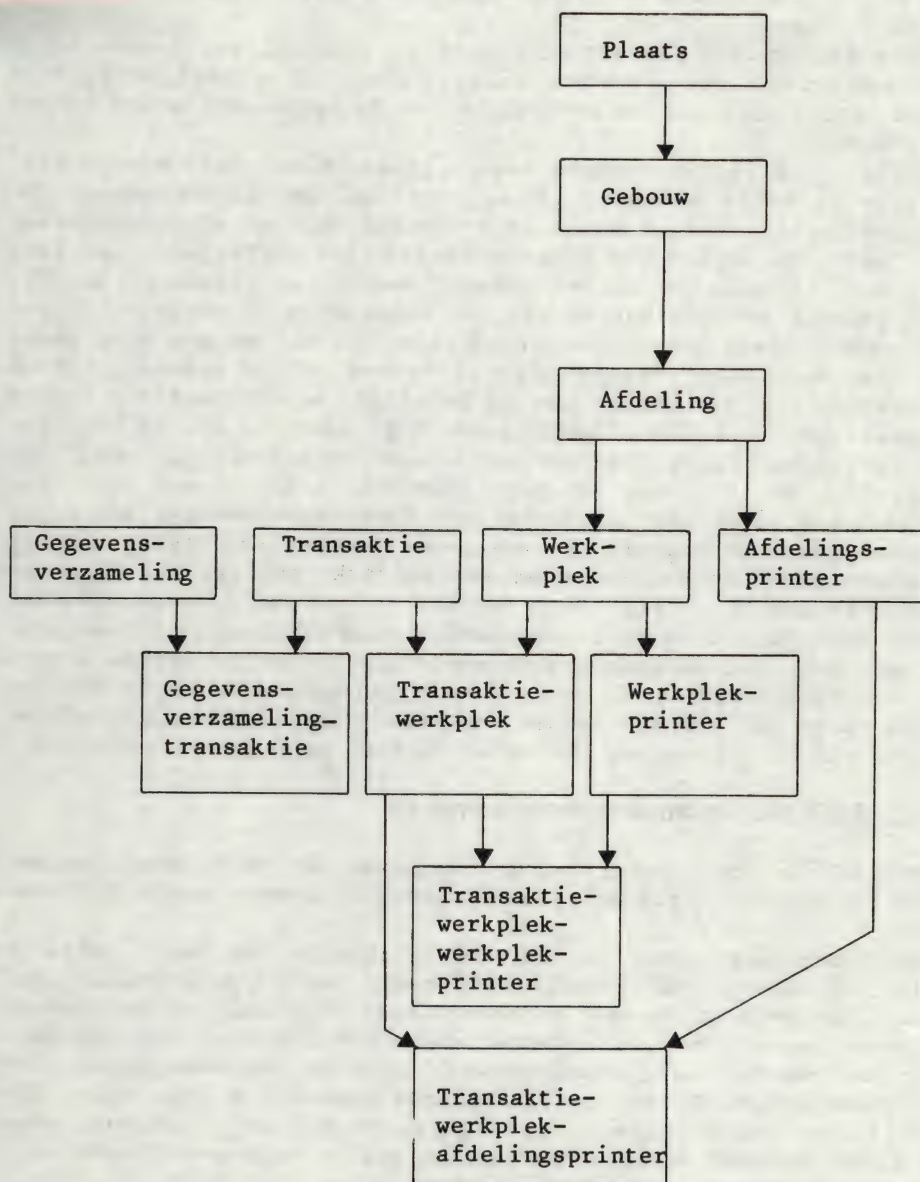


Fig. 54.2 Transakties en geografie als gegevensmodel.

- In C2N-omgevingen gaat het om a-synchroon batch-verkeer. Voor rekenwerk aan dat soort situaties gaat het om het aantal records en het aantal tekens per record. Daarin levert Transaktie analyse geen bijdrage.
- Als het in een C5N-omgeving gaat om interaktieve toepassingen en gedistribueerde gegevens, levert Transaktie analyse cijfers voor het verkeer tussen de systemen en de systeembelasting van de systemen.

In Fig. 54.2 is het verband tussen transakties, printers en geografie in beeld gebracht in de vorm van een gegevensmodel. De geografische situatie wordt in kaart gebracht op werkplekniveau. Per werkplek worden een of meer transakties uitgevoerd. Een deel van die transakties zullen gebruik maken van printers. Daarbij kan gebruik gemaakt worden van een printer op de werkplek of van een afdelingsprinter. Per transaktie kan nu worden aangegeven hoeveel regels op elke printer of op een van de printers worden afgedrukt. Op die manier kan de benodigde printkapaciteit worden bepaald. Per transaktie wordt gebruik gemaakt van een of meer gegevensverzamelingen. De entiteit gegevensverzameling maakt een koppeling mogelijk met een gegevensmodel. Zoals in het deel voor de informatie-analist is aangegeven moet de verwerking zoals die op een transaktieschema is aangegeven, worden vergeleken met het gegevensmodel of de bestanden en het funktiemodel. Bij de gegevensverzameling in Fig. 54.2 kan men denken aan zo'n gegevensmodel, maar in het kader van netwerkontwerp in distributieve omgevingen gaat het om gegevensverzamelingen zoals die ergens kunnen worden opgeslagen en via het netwerk getransporteerd naar een lokatie waar een transaktie die er gebruik van maakt, wordt uitgevoerd. In 54.5 komen we dus terug op deze gegevensverzamelingen.

54.2 Informatieplan en netwerkontwerp

Het doel van deze paragraaf is: aangeven wat er tijdens het opstellen van een informatieplan aan netwerkontwerp gedaan kan worden.

Via interviews wordt de informatiebehoefte in kaart gebracht (35). Op deze wijze ontstaat een beeld van de organisatie: het bedrijfsmodel. Er bestaat uiteraard enige vrijheid in de nauwkeurigheid waarmee men de informatiebehoeften in kaart wil brengen. Men kan daarbij zo globaal te werk gaan dat de werkplekken niet in beeld komen. Binnen de funktionele eenheden worden alleen aktiviteiten onderscheiden en de gegevens die erbij gebruikt worden. Het netwerk vormt de verbinding tussen gegevensbehoefte en gegevensopslag.

De topologie van de gegevensbehoefte kan globaal in kaart gebracht worden door aan te geven welke vestigingen welke gegevens

gebruiken. Men kan ook wat verder gaan en het gegevensgebruik per werkplek in kaart brengen. De grootste nauwkeurigheid wordt bereikt door van het logisch ontwerp alvast het transaktie-ontwerp uit te voeren. Of dat haalbaar is, hangt met name af van het aantal transakties. Als er dus tijdens het opzetten van het informatieplan iets gedaan moet worden aan netwerkontwerp kan dat op een aantal manieren die verschillen in nauwkeurigheid.

- Per vestiging wordt in kaart gebracht welke gegevens worden gebruikt, zonder de zaak op werkplekniveau te analyseren. Op grond van de mogelijkheden voor gegevensopslag kan een aantal netwerkplaatjes worden opgesteld. Het is niet verstandig hieruit een kostenplaatje af te leiden.
- Een ervaren informatie-analist stelt samen met de gebruikers globale transaktieschema's op voor een globale Transaktie analyse. De resultaten zijn dus ook globaal, maar nu ontstaat in ieder geval een beeld van het soort netwerk dat nodig zal zijn. Naast het interaktieve verkeer moet dan ook het batch-verkeer in kaart gebracht kunnen worden. Zo ontstaat een indruk van de hoeveelheid verkeer en het gebruik van het netwerk zodat bijvoorbeeld kan worden aangegeven wat vaste en wat geschakelde lijnen zouden kunnen worden, of hoe het kostenplaatje van Datanet 1 er uit zal zien.
- In het deel voor de informatie-analisten is aangegeven dat transaktie-ontwerp tijdens het vooronderzoek kan plaatsvinden en dat dialoogsimulatie soms een bijdrage kan leveren in de analysefase. Kombinatie van beide levert de mogelijkheid bij het opstellen van een informatieplan al te beginnen met het transaktie-ontwerp van de transakties die te maken krijgen met het netwerk. Op die manier kan in een vroeg stadium vrij nauwkeurig iets gedaan worden aan netwerkontwerp. Zeker als het gaat om de keuze van een infrastruktuur is de kwantificering van groot belang. Als het gaat om een beperkt aantal transakties dat op veel plaatsen dagelijks kontinu wordt uitgevoerd is het de moeite waard dit soort analyses zo nauwkeurig mogelijk uit te voeren.

In de praktijk blijkt vaak dat men in dit stadium, waarin nog te weinig gegevens beschikbaar zijn, een indruk wil hebben van de kosten. Met veel nattevingerwerk ontstaan vaak plaatjes waaraan ontwerpers zich later moeten houden, zeker wat de kosten betreft. Daarmee is dan weer een keer het argument geleverd voor te weinig geld en te weinig tijd tijdens de projekten. Wie bruikbare plaatjes wil hebben van de te verwachten kosten, moet voldoende tijd en geld beschikbaar stellen om de benodigde gegevens te verzamelen. Met transaktie-ontwerp kan dat en is de bestede tijd niet weggegooid omdat er dan meteen een bruikbaar deel van het logisch ontwerp is uitgevoerd.

54.3 Netwerkontwerp en verkeer

- Inleiding.

Van een transaktie-analist wordt verwacht dat hij iets van datakommunikatie afweet. De verkeersresultaten van Transaktie analyse moeten immers vertaald worden naar een netwerkontwerp. In een eerste plaats dient basiskennis aanwezig te zijn op het niveau van (36). Daar worden alle aspekten van datakommunikatie behandeld met een diepgang die verdere studie mogelijk maakt. In de tweede plaats moet de transaktie-analist de datakommunikatie-aspekten bestuderen van de systemen waar mee hij in aanraking komt. In het vervolg van deze paragraaf gaan we er van uit dat de basiskennis aanwezig is. Vaktermen worden in de verklarende woordenlijst uitgelegd. Tenslotte nog een opmerking voor D.C.-specialisten, we gaan in deze paragraaf voorbij aan allerlei details, het gaat om het principe. Bij de lijnsnelheid bijvoorbeeld, gaan we voorbij aan het verschil tussen baud en BPS.

In de inleidende paragrafen is al aangegeven dat het in dit boek gaat om de invulling van twee witte vlekken. Een van die vlekken bevindt zich tussen de systeemontwikkeling en het netwerkontwerp. Het eigenlijke netwerkontwerp is al talloze malen beschreven, alleen zonder aan te geven hoe het benodigde cijfermateriaal moet worden verkregen. Transaktie analyse levert, afgeleid van de kwalitatieve en kwantitatieve eisen van de gebruikers, de cijfers die alle netwerkontwerpers nodig blijken te hebben: de berichtlengtes en het aantal berichten per tijdseenheid, het omgekeerde van de invoerrepetitietijd. Het is niet de bedoeling, die boeken te herschrijven, we zullen alleen de aansluiting tussen de cijfers van Transaktie analyse en het netwerkontwerp beschrijven.

- Wat is verkeer?

Het verkeer in een netwerk is het aantal tekens dat per tijdseenheid getransporteerd wordt. Vaak worden tekens in blokken verstuurd. In dat geval wordt het verkeer dus bepaald door het aantal blokken of berichten per tijdseenheid en het aantal tekens per blok. Een netwerk bestaat uit een of meer datakommunikatieverbindingen of -lijnen. Iedere lijn heeft een vaste snelheid waarmee tekens worden getransporteerd. De lijnsnelheid zou dus kunnen worden uitgedrukt in tekens per sekonde. Omdat het aantal bits per teken nogal kan verschillen wordt de snelheid van de lijn uitgedrukt in bits per sekonde (BPS). Een lijn van 2400 BPS kan dus maximaal 2400 bits per sekonde transporteren. Is een teken samengesteld uit 8 bits, dan kan die lijn dus 300 tekens per sekonde transporteren. Als ieder bericht bestaat uit 100 tekens kan die lijn 3 berichten per sekonde transporteren. Het transporteren van berichten verloopt meestal volgens een bepaalde procedure. Een standaardprocedure is bijvoorbeeld:

- vragen of de geadresseerde een bericht kan ontvangen,
- positief reageren op de vraag,
- versturen van een of meer berichten,
- wachten op de bevestiging van goede aankomst van de berichten,
- eventueel opnieuw versturen van berichten,
- afsluiten van de verbinding.

De overhead betekent extra tekens en tijd. De procedure-overhead (POV) is per soort procedure anders. Voorbeelden van procedures zijn: BSC, HDLC en SDLC. Deze procedures zijn al talloze keren beschreven in de vakliteratuur.

- Welke soorten verkeer kunnen we onderscheiden?

In het kader van dit hoofdstuk behandelen we drie soorten verkeer: batch-verkeer, interaktief verkeer en interaktief batch-verkeer. Bij batch-verkeer worden door een computer of een terminal records verstuurd. De lijnsnelheid vormt het knelpunt in het transport. Het bestand wordt in blokken verstuurd. Per blok wordt een aantal besturingstekens toegevoegd en tussen twee blokken gaat enige tijd verloren bijvoorbeeld om de goede aankomst van een blok te bevestigen. Deze POV verschilt per lijnprocedure. Een lijn van 2400 BPS kan dus in een uur geen 3600x2400/8 tekens van 8 bits transporteren. Men spreekt in dit verband van de effektieve lijnsnelheid die als volgt wordt berekend:

$$\frac{I(M-C)}{\frac{M}{R} + T} \text{ bits per sekonde}$$

I: aantal bits per teken

M: aantal tekens per blok

C: aantal besturingstekens per blok

R: lijnsnelheid in tekens per sekonde

T: tijd tussen de blokken in sekonden

Bij lijnsnelheden tot 2400 BPS is de effektieve lijnsnelheid meestal 80-90% van de nominale, bij 9600 BPS kan dat percentage afnemen tot 75%. Als bij koppelingen tussen computers een van beide systemen het knelpunt vormt, bijvoorbeeld omdat T groot is, neemt het percentage nog verder af. Tenslotte kan de effektieve lijnsnelheid nog lager worden doordat bij lijnstoring af en toe een blok opnieuw verstuurd moet worden. De foutkans moet in dat geval worden omgerekend naar de tijd tussen de blokken (36).
Om een indruk te krijgen van de tijd die een transport duurt zonder rekening te houden met alle details, wordt vaak gezegd: de transporttijd = 10 x M/lijnsnelheid in BPS.
Interaktief verkeer is er om tekens van een beeldscherm naar een computer te sturen en terug. Dit verkeer is een direkt gevolg van de dialoog. Vraag en antwoord leiden tot een bericht heen en een bericht terug. De tijd tussen twee interakties hangt af van de procedure aan het beeldscherm. De gemiddelde tijd tussen twee interakties is de invoerrepetitietijd.
Het verkeer tussen twee computers heeft hetzelfde karakter, als gedurende een interaktie een programma gegevens opvraagt bij een andere computer. Als vraag of antwoord bestaat uit een aantal blokken die als batch worden verstuurd spreken we van interaktief batch-verkeer. Dit verkeer is meestal slechts indirekt afhankelijk van de dialoog aan het beeldscherm, omdat het gaat om de kommunikatie tussen twee applikaties ten dienste van die dialoog.
Ook voor interaktief verkeer geldt een bepaalde lijnprocedure die moet worden vastgesteld. In Fig. 54.3 is:

- t1: de tijd die nodig is om de kommunikatie te starten: een halve pollcyclus, een ENQ-ACK-kombinatie of iets anders
- t2: de tijd die nodig is voor het transport van de besturingstekens als STX, ETX, BCC, PAD, frame header, frame tail, transportgegevens.
- t3: de tijd die nodig is om het bericht te transporteren
- t4: de tijd die nodig is om de ontvangst te bevestigen: ACK, frame of iets anders
- t5: de tijd voor de afsluiting van de kommunikatie: EOT, frame blok of iets dergelijks. Ongeveer een responsetijd later komt het bericht terug waarvoor dezelfde regels gelden. Wanneer men na afloop van de invoerrepetitietijd weer op de ENTER-toets drukt herhaalt zich het bovenstaande.

Het zal duidelijk zijn dat de hoeveelheid verkeer uitgedrukt in tekens per tijdseenheid of berichten per tijdseenheid bepaald wordt door berichtlengte en invoerrepetitietijd. De berichtlengte wordt bepaald door de dialoog en het soort beeldscherm, de invoerrepetitietijd door de procedure aan het beeldscherm. Dat is de reden waarom men in de vakliteratuur altijd uitgaat van voorbeelden: laten we eens aannemen dat er 120 berichten per uur wor-

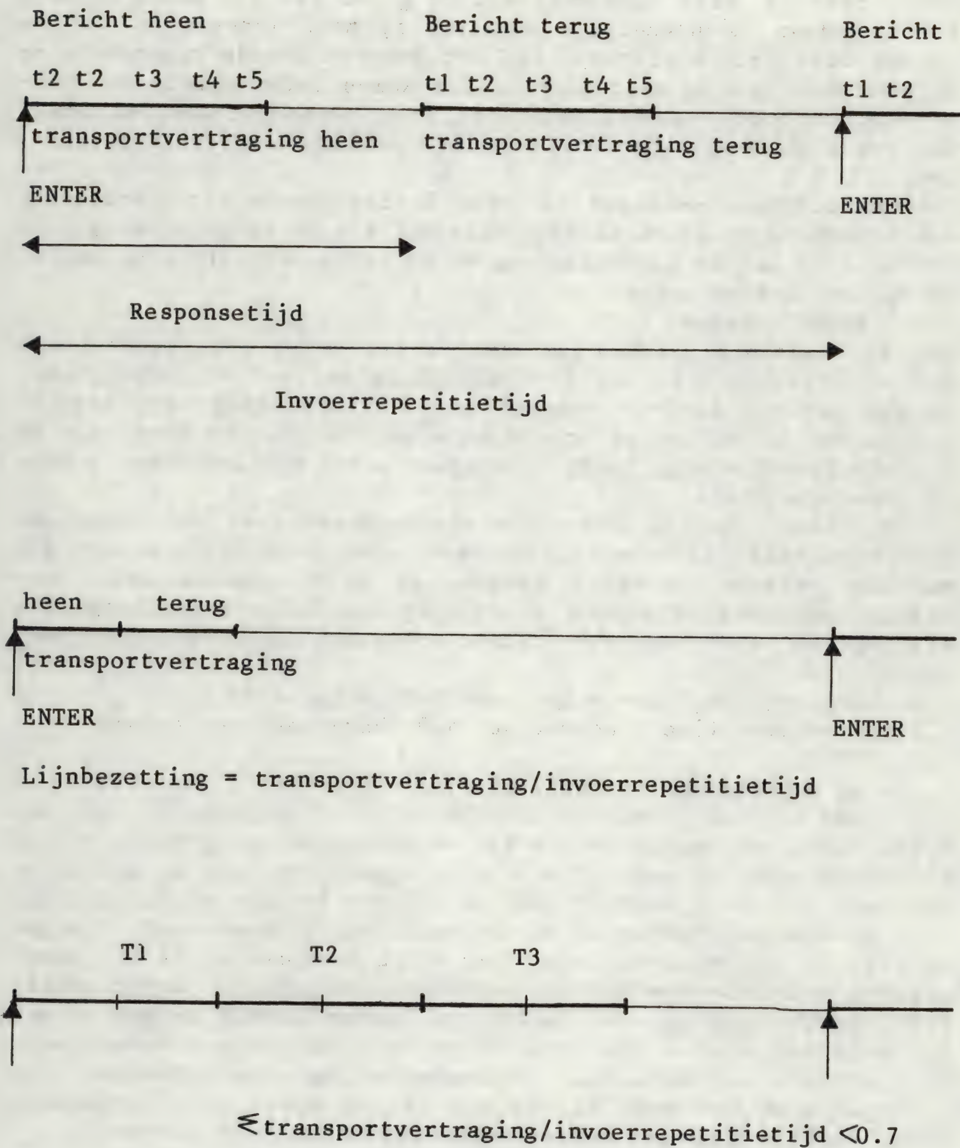


Fig. 54.3 Bepaling van de lijnbelasting

den verstuurd. Alle rekenwerk dat op grond van dit soort cijfers wordt gedaan is natuurlijk uitstekend, maar Transaktie analyse levert die cijfers uitgaand van het ontwerp van de transaktie op de werkplek van de gebruiker. Zover komen leveranciers van netwerken niet. De verkeersresultaten van Transaktie analyse maken het mogelijk de gevolgen van het interaktieve verkeer te berekenen.
Interaktief batch-verkeer is voor de datakommunikatieverbinding batch-verkeer en de berekening verloopt als is aangegeven. Bij de behandeling van de lijnbelasting en de transportvertraging komen we op dit verkeer terug.
- Het soort terminal
Het is duidelijk dat het verkeer in een netwerk te maken heeft met de intelligentie van de terminal en met het dialoogontwerp. Om een netwerk door te rekenen zijn cijfers nodig. Die cijfers zouden dus af te leiden moeten zijn uit de gekozen terminals en de ontworpen dialoog. In de literatuur wordt het probleem op twee manieren opgelost.
In (8) staan 6 hoofdstukken over dialoogontwerp en 5 hoofdstukken over terminals. In deze hoofdstukken komen geen cijfers voor die met het verkeer te maken hebben. In de 17 hoofdstukken over Design calculations worden de cijfers die het verkeer betreffen als gegeven beschouwd. Alle rekenvoorbeelden beginnen dan ook met zinnen als:
- We have estimated the mean responsetime (p. 439)
- A half-duplex line transmit at 14.8 characters per second (p. 555)
- Let us suppose ... (p. 428)
- A study ... has given the distributions of lengthes ... (p.400)
Medewerkers van Martin zeggen dat ze die gegevens opvragen bij de bedrijven waar ze een netwerk voor ontwerpen. Als de automatiseerders van zo'n bedrijf (8) bestudeerd hebben om die cijfers vast te stellen, hebben ze nu een probleem: niemand heeft die gegevens. De andere manier die veel wordt beschreven, is het doorrekenen van netwerken met simulatiemodellen. In (18) wordt uitgebreid beschreven hoe zo'n model zou kunnen werken en wat er mee berekend kan worden. Als de cijfers over het verkeer beschikbaar zijn, kunnen modellen zeer interessante gegevens opleveren. Hoe belangwekkend het boek verder ook is, er wordt niet aangegeven hoe die cijfers moeten worden bepaald. Als er wordt gerekend, gaat dat met willekeurig aangenomen cijfers.
Karakteristieke zinnen in dit verband zijn:
- All you need to know is the mean input rate (p 63)
- For example, there are 3 packets/sec. (p 63)
- Let us now assume a mean packet size ... (p 64)
- An important question is how much traffic a network can support

(p 65)
Het is natuurlijk interessant om te berekenen hoeveel verkeer een netwerk kan verwerken onder allerlei omstandigheden. Toch is het antwoord op die vraag pas van belang als bekend is welke kapaciteit er nodig is. Wat er nodig is wordt aangegeven door verkeersresultaten van Transaktie analyse. Die resultaten ontstaan doordat op het detailschema zowel de dialoog als de intelligentie van de terminal worden vastgelegd.
In (36) wordt een aantal voorbeelden gegeven van de hoeveelheid tekens die naar beeldschermen met verschillende intelligentie moet worden gestuurd om dezelfde dialoog te realiseren.
Er zijn vaak een aantal manieren om een terminal met een gegeven intelligentie, te besturen. Met name bij de eenvoudige start/stop terminals kan de ontwerper uit een aantal mogelijkheden kiezen. Hij kan bijvoorbeeld, onafhankelijk van het aantal tekens op een regel, altijd regels van 80 tekens oversturen. Hij kan echter ook optimaliseren op het aantal tekens en alleen de noodzakelijke tekens oversturen.
In dit verband kan nog worden opgemerkt dat soms die besturing vastligt in het operating system. Bij sommige systemen wordt per interaktie altijd een nieuw masker verzonden. Met name bij software gemaakt met vierde-generatietalen is dit vaak het geval. Dat heeft zoveel konsekwenties voor het verkeer, dat in dat soort omgevingen even uitgezocht moet worden hoe de besturing van het beeldscherm gerealiseerd is. De transaktie-analist moet voldoende op de hoogte zijn van de werking van de betrokken beeldschermen om het detailschema goed te kunnen invullen.
- Wat is de lijnbelasting?
De lijnbelasting is de verhouding tussen de tijd die de lijn gebruikt om tekens te transporteren en de beschikbare tijd. Bij batch-verkeer is gedurende het transport van de blokken de lijn kontinu bezet, behalve gedurende de tijd tussen de blokken. Als die tijd echter een gevolg is van de lijnprocedure betekent dat, dat de lijn voor 100% bezet is voor het transporteren van het bestand. Als het bestand record voor record zou moeten worden overgestuurd, terwijl tussen twee records de lijn vrijgegeven moet worden voor ander verkeer, is flow control nodig van een hoger niveau dan de lijnprocedure. Zo kan bijvoorbeeld het 2780-protocol of het 3270-protocol gebruikt worden om, onder besturing van een aplikatie, een batch van records te versturen van een mainframe naar een micro.
In het algemeen gaan we echter uit van batch-verkeer dat een lijn voor 100% bezet. Bij interaktief verkeer is de lijnbezetting per beeldscherm gelijk aan de tijden zoals die zijn aangegeven in Fig 54.3, gedeeld door de invoerrepetitietijd. Anders gezegd: (de transportvertraging heen + de transportvertraging terug)/ de

invoerrepetitietijd, of: de transportvertraging / de invoerrepetitietijd, zoals die in het midden van Fig. 54.3 getekend. Na afloop van de transportvertraging wordt de lijn niet gebruikt, totdat, bij de volgende ENTER, de lijn weer wordt benut gedurende de transportvertraging. Daarmee is de lijnbelasting bekend. Hoe kleiner de lijnbelasting per terminal, hoe meer beeldschermen kunnen worden aangesloten. In Fig. 54.3 zouden, als de verhoudingen zouden kloppen, drie beeldschermen kunnen worden aangesloten. Zoals bekend is uit de wachtrijtheorie mag een server maximaal voor 70% worden belast. De server is de lijn, de servicetijd de transportvertraging en de aankomstdichtheid het omgekeerde van de invoerrepetitietijd. Omdat Transaktie analyse naast de gemiddelde waarden, ook de varianties levert zijn alle gegevens beschikbaar voor 95%-situaties, worst cases etc.
In de praktijk zal het niet zo vaak gebeuren dat er per cluster altijd dezelfde transakties worden uitgevoerd. Ook ten aanzien van de bepaling van het verkeer bestaan de twee eerder genoemde mogelijkheden om dit soort situaties door te rekenen: via combitransakties of via het bezettingsoverzicht van de ergonomische Transaktie analyse. Als de gebruikers kontinu verschillende transakties door elkaar gebruiken, kan het beste met kombitransakties gewerkt worden. Dan ontstaat voor alle betrokken beeldschermen een gewogen gemiddelde transaktie, met een transportvertraging en een invoerrepetitietijd. Als af en toe van transaktie gewisseld wordt, maar men toch enige uren achter elkaar met eenzelfde transaktie bezig is kan de worst case van de lijnbelasting worden bepaald, met behulp van het tijdbestedingsdiagram.
Dat brengt ons bij het laatste element in de lijnbelasting: de lijnsnelheid. We hebben de lijnbelasting omschreven als de transportvertraging/de invoerrepetitietijd. De transportvertraging hangt van twee zaken af: het aantal tekens en de snelheid waarmee die tekens kunnen worden vervoerd. De tijden t2, t3, t4 en t5 worden dus berekend door het aantal tekens te vermenigvuldigen met het aantal bits per teken en te delen door de lijnsnelheid in BPS. Het aantal tekens dat betrokken is bij de berekening van t1, t2, t4 en t5 hangt af van de lijnprocedure, het aantal tekens dat gebruikt wordt bij de berekening van t3 is de gemiddelde berichtlengte van Transaktie analyse. Nu kan dus, gegeven het aantal beeldschermen, de berichtlengte en de invoerrepetitietijd, berekend worden welke lijnsnelheid nodig is om het verkeer te realiseren met een lijnbelasting kleiner dan 70%.
Dat geldt zowel voor lijnen die aansluiting geven op een eigen netwerk, als voor lijnen die aansluiting geven op Datanet 1. In het laatste geval moeten alleen de berichtlengtes even worden omgerekend naar pakketten. De pakketlay-out is bekend en de lijnprocedure is HDLC. Er gaan maximaal 128 tekens in een pakket. Zie

paragraaf Transaktie analyse en Datanet 1. Dan tenslotte nog iets over interaktief batch-verkeer. In feite gaat het, wat het verkeer betreft dan meestal om een kombinatie van interaktief en batch-verkeer. Omdat het berichtenverkeer van de interaktieve toepassingen bekend is, kan ook worden uitgerekend wat er gebeurt als er via dezelfde lijn ook nog print- of micro-files worden overgestuurd.
Bij alle kombinaties van batch- en interaktief verkeer is het van belang goed uit te zoeken hoe het batch-verkeer bestuurd kan worden. Als alle records in een keer worden verstuurd zijn alle beeldschermen enige tijd geblokkeerd. Als de records een voor een worden getransporteerd kunnen de beeldschermen wel doorwerken, maar met responsetijden waarvan de verlenging berekend kan worden. Als geen van beide akseptabel is, kan overwogen worden een aparte lijn te gebruiken voor batch-verkeer. Als dat verkeer maar een enkele keer per dag plaatsvindt bijvoorbeeld naar een micro, dan kan een geschakelde verbinding gekozen worden. Als micro's worden gebruikt om als back-up te dienen voor het werken met de centrale database, is er een degelijk logisch en technisch ontwerp nodig. Niet alleen om de software te ontwerpen voor het maken en versturen van het extrakt van de database, maar ook voor het netwerk dat er voor nodig is.
Bij koppelingen tussen computers onderling is het niet meer de lijnprocedure die bepaalt of het gaat om interaktief of batchverkeer. De lijnprocedure 3270 BSC is ontwikkeld voor interaktieve kommunikatie tussen een beeldscherm en een computer. De 2780/3780-procedure diende voor het transport van batches, kaarten, printregels of in het algemeen: grote hoeveelheden records. Bij interaktief verkeer is iedereen daarom geneigd aan 3270 BSC te denken of aan vergelijkbare procedures. Als bij verbindingen tussen computers toepassingen met elkaar kommuniceren, kan een batch van printregels worden verstuurd via de 3270-procedure, maar er bestaan talloze interaktieve koppelingen dia via 2780-BSC een bericht heen en een bericht terug sturen.
Voor het versturen van een file is de 2780-procedure het meest efficient, maar vaak niet te besturen door een applikatie, terwijl de 3270-procedure dat wel is. Dat betekent dat bij een gedwongen kombinatie van batch- en interaktief verkeer via de 3270-procedure het batch-verkeer niet de hele verbinding voor de duur van het transport onbruikbaar mag maken voor interaktief verkeer. Het ontvangende station stuurt bijvoorbeeld een getimede ENTER om de volgende groep regels in de schermbuffer te aksepteren.
Kortom, in distributieve omgevingen gaat het om interaktief verkeer als het transport deel uitmaakt van de responsetijd. Het kan dan gaan om interaktief verkeer bestaand uit een bericht heen en een bericht terug via welke lijnprocedure dan ook, of om interak-

tief batch-verkeer waarbij een aantal records wordt uitgewisseld via een willekeurige lijnprocedure. De aanduiding interaktief batch-verkeer heeft alleen tot doel iedereen te attenderen op een mogelijk probleem met de responsetijd: er moet een batch worden aangemaakt en worden getransporteerd.

- Transportvertraging en responsetijd.

Tot nu toe is het begrip transportvertraging alleen genoemd in verband met de lijnbezetting. Bij gegeven berichtlengtes en aantal beeldschermen moet de lijnsnelheid zodanig worden gekozen dat de lijn niet overbelast raakt. Als op die manier de lijnsnelheid is bepaald, ondervindt het verkeer per ENTER die transportvertraging. Dat element van de responsetijd is daarmee bepaald. Aan het eind van het technisch ontwerp, als de responsetijden voor het laatst worden bezien, kan dit aspekt worden meegenomen.

Bij interaktief batch-verkeer kan worden berekend hoeveel tijd het kost het bestand over te sturen. Bij kombinaties van batch- en interaktief verkeer kan worden uitgerekend wat de responsetijden worden gedurende het transport van de batch.

54.4 Netwerkontwerp in C1N/C3N-omgevingen

In een C1N/C3N-omgeving is de lokatie van de gegevensopslag bekend. Per vestiging wordt vastgesteld welke transakties er worden uitgevoerd. De verkeersresultaten van Transaktie analyse kunnen op twee manieren in kaart worden gebracht:

- via kombitransakties wordt de gewogen, gemiddelde transaktie bepaald en de hoeveelheid verkeer.
- uit het bezettingsoverzicht van de ergonomische analyse wordt vastgelegd welke transakties op welke werkplekken worden uitgevoerd. Per cluster van beeldschermen wordt nu vastgesteld welke kombinatie van transakties het meeste verkeer oplevert. Voor die hoeveelheid verkeer wordt een lijn gekozen die, bijvoorbeeld in 95% van de gevallen, een lijnbelasting oplevert die kleiner is dan 70%.

In het geval van een kombinatie van batch- en interaktief verkeer moeten de transportvertraging en de lijnbelasting worden berekend voor het interaktieve verkeer bij afwezigheid van en gedurende het batch-verkeer. Er moet worden vastgesteld hoe de flow van het batch-verkeer wordt geregeld. Bij printers moet worden vastgesteld hoe de printregels worden verstuurd ten opzichte van het interaktieve verkeer. Als per pollcyclus bijvoorbeeld een printregel wordt verstuurd betekent dat dat de pollcyclus verlengd wordt met de tijd die nodig is om een regel te transporteren en dus kan de nieuwe POV worden berekend. Voor de lijnbelasting betekent deze implemetatie van het batch-verkeer, dat per invoerrepetitietijd een aantal regels wordt getransporteerd gelijk aan de

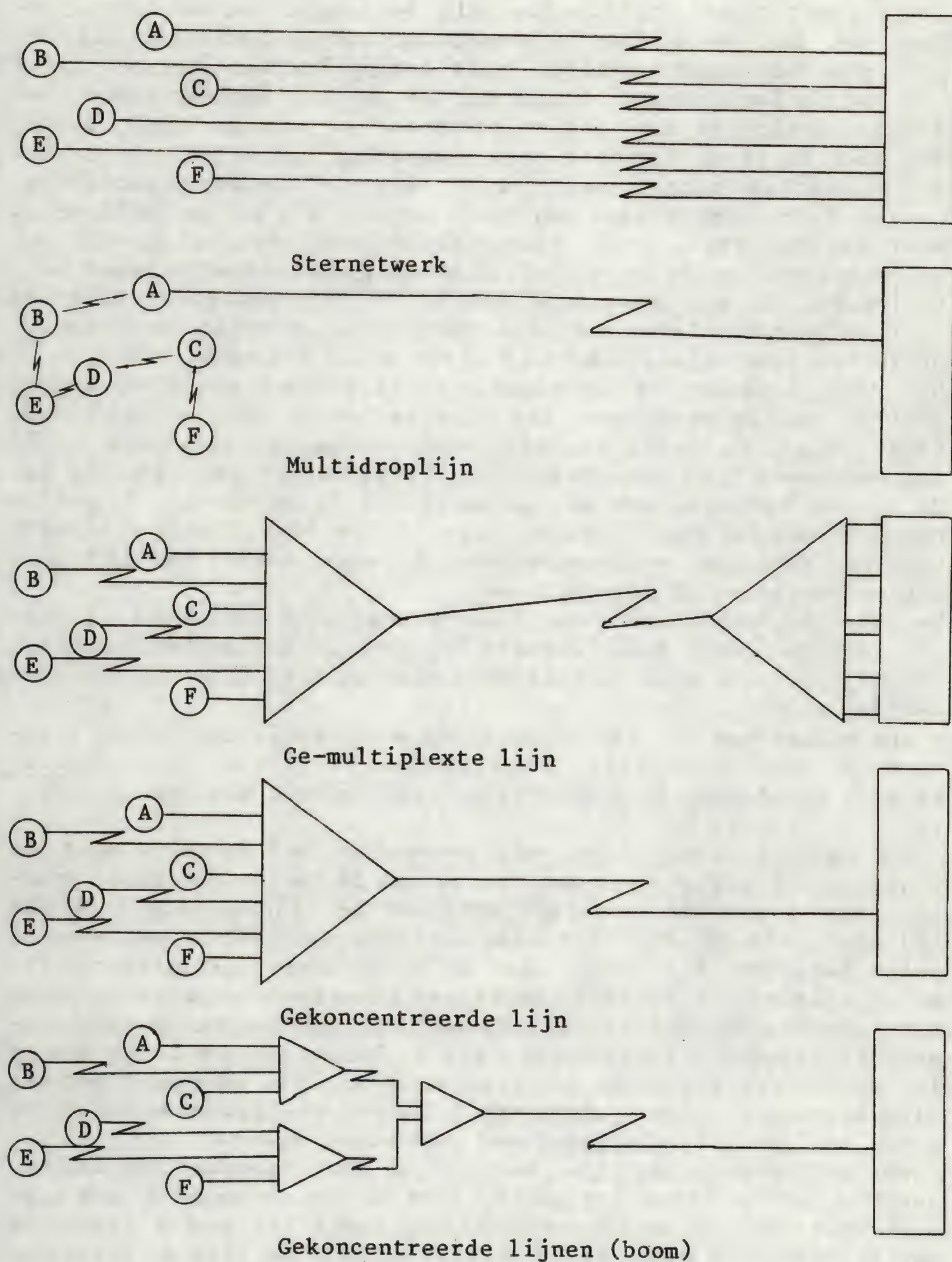


Fig. 54.4 Van ster naar boom.

invoerrepetitietijd/pollcyclus. Als de lengte van de regels bekend is, kan de nieuwe lijnbezetting worden berekend. Bij een goed gedimensioneerd systeem duurt een pollcyclus maar een fraktie van de invoerrepetitietijd en dus zal het verkeer enorm toenemen. Op zich is dat voor de responsetijd geen probleem, als de POV niet te sterk toeneemt. Als tengevolge van het batch-verkeer de pollcyclus vergelijkbaar wordt met de invoerrepetitietijd, worden de responsetijden merkbaar langer. Als er per pollcyclus meer dan een regel wordt overgestuurd wordt de invloed van het batch-verkeer op de responsetijden nog groter. Statistische multiplexers die op dynamische manier de lijnkapaciteit verdelen over de poorten, laten het hier afweten. In principe is de kombinatie van interaktief- en batch-verkeer uit den boze.
Het netwerkontwerp in ClN-omgevingen begint met een sternetwerk: iedere lokatie wordt met een lijn verbonden met het mainframe (Fig. 54.4). Op grond van het berekende aantal terminals en de verkeersresultaten van Transaktie analyse wordt de belasting van de lijnen berekend met als parameter de lijnsnelheid. Aangezien er maar weinig mogelijkheden zijn, is de hoeveelheid rekenwerk beperkt. Nu kunnen de netwerkkosten berekend worden aan de hand van modemprijzen en PTT-tarieven.
Vervolgens worden een paar boomstrukturen doorgerekend. Lijnen die voor een groot deel "parallel" lopen, worden gekombineerd tot een lijn met een paar vertakkingen aan het eind (Fig. 54.4). Voorbeelden zijn:
- een multidroplijn. Per aansluiting wordt de lijnbelasting berekend. De POV is een halve pollcyclus en de tijd om besturingstekens te versturen. De totale lijnbelasting moet kleiner zijn dan 70%
- een gemultiplexte lijn. Met eenvoudige multiplexers moet de lijnsnelheid gelijk zijn aan de som van de snelheden van de poorten. Bij 8 poorten van 1200 BPS moet de lijnsnelheid 9600 BPS bedragen. Als de 1200 BPS aansluitingen voor 5% belast worden, geldt hetzelfde percentage voor de lijn tussen multiplexers. Als de lijnkapaciteit beter verdeeld zou kunnen worden over de poorten, zou men met een lagere lijnsnelheid toe kunnen. Hoeveel lager, is eenvoudig te berekenen als de bezetting per poort bekend is. De verkeerresultaten van Transaktie analyse maken die berekening eenvoudig. Leveranciers van dit soort multiplexers zullen op grond van deze cijfers graag een aanbieding maken.
- een gekoncentreerde lijn. Nu wordt op het knooppunt van de takken een koncentrator geplaatst. Voor de dimensionering van koncentrators hebben de leveranciers per poort het aantal berichten per tijdseenheid nodig en de berichtlengte. Dat zijn de verkeersresultaten van Transaktie analyse. Natuurlijk kan nog worden overwogen koncentrators in serie te zetten om meer vertakkingen

te maken en minder poorten van het mainframe te gebruiken. Leveranciers van netwerken kunnen verschillende oplossingen nu precies doorrekenen. Dat is immers hun vak. Het probleem was altijd dat ze geen gegevens hadden over het verkeer. Die zijn nu beschikbaar.

54.5 Netwerkontwerp in distributieve omgevingen

In distributieve omgevingen is een netwerk nodig om het geografische verschil tussen gegevensgebruik en gegevensopslag goed te maken. Er kan onderscheid gemaakt worden tussen direkt verkeer, dat nodig is om gegevens te transporteren van de opslagplaats naar de lokatie waar ze gebruikt worden en indirekt verkeer, dat nodig is om de konsistentie van de gegevens te verzorgen.
De soorten verkeer kunnen ook op een andere manier worden ingedeeld:
- batch-verkeer, dat onafhankelijk van transakties plaats heeft (Fig. 54.5 tot .7),
- interaktief verkeer, waarbij het gaat om een bericht heen en een bericht terug (Fig. 54.8 tot .10),
- interaktief batch-verkeer, wat meestal een bericht heen en een bestand terug betekent. Karakteristiek voor interaktief (batch-) verkeer is dat het deel uitmaakt van de responsetijd en voorkomt op het transaktieschema. In Fig. 54.5 gaat het kennelijk om een transaktie die hoogstens een paar keer per dag wordt uitgevoerd. Zou deze figuur het begin van een order entry-transaktie zijn, die vele malen per dag wordt uitgevoerd, dan ging het om interaktief batch-verkeer.
Zowel het direkte als het indirekte verkeer kan in een van deze drie vormen plaatsvinden.
In (7) wordt een zestal mogelijkheden besproken voor gegevensdistributie: (39)
- Replicated data: de verschillende lokaties hebben elk een kopie van de centrale databank (redundant gegevensopslag) waarbij de up-dates van de lokale kopiebestanden uiteraard goed bewaakt moeten worden.
- Subset data: lokaties beheren (disjuncte) subsets van de centrale databank voor data entry, retrieval of lokatie gegevensverwerking.
- Reorganized data: operationele gegevens worden vanuit de operationele data bases geaggregeerd en gehergroepeerd voor data bases ten behoeve van Decision Support Systemen of andere informatiesystemen.
- Partioned data: lokaties gebruiken elk eenzelfde database qua struktuur, maar niet dezelfde occurrences.
- Separate-scheme data: lokaties gebruiken elk data bases met een

verschillende gegevensstruktuur, maar wel passend in een totaalstruktuur (deelschema's in de CODASYL terminologie).
- Incompatible data: de verschillende lokaties gebruiken puur lokale gegevens, die op geen enkele wijze binnen andere lokaties gebruikt (kunnen) worden.
Bij de eerste 3 wordt in (7) onderscheid gemaakt tussen de synchrone en de a-synchrone vorm. Synchroon betekent in dit verband dat de gegevens op alle lokaties op elk moment konsistent moeten zijn. Dat betekent updates per muterende transakties en dus interaktief (batch-)verkeer dat in kaart gebracht kan worden op decentrale transaktieschema's. Het batch-verkeer hangt alleen af van de hoeveelheid records die moet worden verstuurd. Zowel het direkte als het indirekte verkeer hebben meestal niets te maken met het dialoogontwerp. Het gaat om verkeer tussen applikaties. Hoogstens kan uit de dialoog worden afgeleid welke gegevens minstens opgehaald moeten worden. Voor de applikatie-ontwerper maakt het weinig uit of een vastgestelde hoeveelheid gegevens van de schijf of van de lijn komt.
Het direkte verkeer wordt in kaart gebracht via decentrale transaktieschema's. Bij synchrone gedistribueerde databases wordt ook het indirekte verkeer op het decentrale transaktieschema aangegeven. Nu de soorten verkeer in distributieve omgevingen zijn be-

TRANSAKTIESCHEMA decentraal
Transaktienaam: UPDATE VOORR.BEST.

Menselijke handelingen en bewerk.		Decentrale verwerking		Centrale verwerking
Via Hoofdmenuscherm starten van update bestand VOORRAAD	---)	Start update progr. op centr. systeem	---)	Start update programma, oversturen van n records
		lezen records, voorraadbest. updaten. Displayen info over verloop van transport	(---	
Kontrole op goede afloop	(---			

Fig. 54.5 Voorbeeld van batch-verkeer.

TRANSAKTIESCHEMA decentraal
Transaktienaam: UPDATE VOORR.BEST."AUTOMATISCH", DECENTRAAL INITIATIEF

Menselijke handelingen en bewerkingen		Decentrale verwerking		Centrale verwerking
		Om 19.00 uur starten van upd. progr. op centr. systeem	---)	Start update programma, oversturen van n records. Info loggen over afloop.
		lezen records, voorraadbest. updaten. Info over afloop loggen.	(---	

Fig. 54.6 Voorbeeld van batch-verkeer.

TRANSAKTIESCHEMA decentraal
Transaktienaam: UPDATE VOORR.BEST. "AUTOMATISCH", CENTRAAL INITIATIEF

Menselijke handelingen en bewerkingen		Decentrale verwerking		Centrale verwerking
				Om 19.00 uur starten van upd. progr. Oversturen van n records
		lezen van n records, updaten voorraadbest. Info loggen over afloop	(---	Info over afloop loggen.

Fig. 54.7 Voorbeeld van batch-verkeer.

TRANSAKTIESCHEMA centraal
Transaktienaam: ORDER INFO

Menselijke handelingen en bewerkingen	Transport	Machinale verwerking
Bepalen van het zoekargument van de klant. Intypen van het zoekargument.		
	----)	Opzoeken klantgegevens. Displayen: - Naam - Adres - Woonplaats - Klantnummer - Status
	(----	
Kontrole van de gegevens Intypen: Kredietgegevens JA/NEE		
	----)	Indien NEE cursor naar eerste artikelregel. Indien JA opzoek kredietgegevens. Displayen: - kredietkode - saldo
	(----	
Kontrole saldo. Intypen van aantal en artikelnummer.		
	----)	Opzoeken en displayen: - artikelomschrijving - aantal in voorraad
	(----	
Kontrole artikelomschrijving Etc.		

Fig. 54.8 Het transaktieschema van de informatie-analist.

TRANSAKTIESCHEMA decentraal
Transaktienaam: ODER INFO 1
DATA-DISTRIBUTIE: KLANT GEG : lokaal
KREDIET & ART.GEG: centr.

Menselijke handelingen en bewerkingen		Decentrale verwerking		Centrale verwerking
Bepalen van zoekargument van klant. Intypen zoekargument.	---)	Opzoeken klantgeg. Displayen van - Naam - Adres - Woonplaats - Klant.nr.		
Kontrole van geg.	(---	- Status		
Intypen: Kredietgegevens JA/NEE	---)	Indien NEE cursor naar le artikel regel. Indien JA kred. geg. centraal opvragen via klant.nr.	---)	Opzoeken en versturen van kredietgeg. displayen - kredietkode - saldo
		Displayen van - kredietkode	(---	
Kontrole saldo	(---	- saldo		
Intypen van aantal en art.nr.	---)	Opvragen bij centraal systeem	---)	Opzoeken en versturen van - art.omschr. - aantal in voorraad
		Displayen art. omschr. vergelijken aantal met	(---	
Kontrole van omschr.	(---	aantal in voorr.		

Fig. 54.9 Implementatie-voorbeeld 1 van Fig 54.8.

TRANSAKTIESCHEMA decentraal

Transaktienaam: ODER INFO 2

DATA-DISTRIBUTIE: - klantgeg + art.omschr: lokaal
- krediet, voorraad: centraal

Menselijke handelingen en bewerkingen		Decentrale verwerking		Centrale verwerking
Bepalen van zoekargument van klant. Intypen zoekargument.	---)	Opzoeken klantgeg. Displayen van - Naam - Adres - Woonplaats - Klant.nr.		
Kontrole van geg.	(---	- Status		
Intypen: Kredietgegevens JA/NEE	---)	Indien NEE cursor naar le artikel regel. Indien JA kred. geg. centraal opvragen via klant.nr.	---)	Opzoeken en versturen van kredietgeg. - kredietkode
		Displayen van - kredietkode	(---	- saldo
Kontrole saldo	(---	- saldo		
Intypen van aantal en art.nr.	---)	Opzoeken art. omschr. en		
Kontrole van omschrijving	(---	displayen		
	---)	Opvragen voorraad bij centrale: art.nr.	---)	Opzoeken voorraad en
		Displayen	(---	versturen

Fig. 54.10 Implementatie-voorbeeld 2 van Fig. 54.8.

TRANSAKTIESCHEMA decentraal

Transaktienaam: ODER INFO 3

DATA-DISTRIBUTIE: Klantgeg., voorraad geg., art.geg.: lokaal
Krediet geg. : centraal

Menselijke handelingen en bewerkingen		Decentrale verwerking		Centrale verwerking
Bepalen van zoek- argument van klant. Intypen zoekargument.	---)	Opzoeken klantgeg. Displayen van - Naam - Adres - Woonplaats - Klant.nr.		
Kontrole van geg.	(---	- Status		
Intypen: Krediet- gegevens JA/NEE	---)	Indien NEE cursor naar le artikel regel. Indien JA kred. geg. centraal opvragen via klant.nr.	---)	Opzoeken en versturen van kredietgeg. - kredietkode - saldo
		Displayen van - kredietkode	(---	
Kontrole saldo	(---	- saldo		
Intypen van aantal en art.nr.	---)	Opzoeken - omschrijving - aantal in voorr. Displayen omschr. Vergelijken aant. met aant. in voorr.		
	(---			

Fig. 54.11 Implementatie-voorbeeld 3 van Fig. 54.8.

sproken kunnen we de aanpak voor het netwerkontwerp in grote lijnen aangeven (zie ook Fig. 37.10).

- Bepaal welke gegevensverzamelingen per vestiging worden gebruikt aan de hand van de transakties die per werkplek worden uitgevoerd.

In (7) worden matrices getekend (Vol II p. 391 en 399) waaruit het verband blijkt tussen PROCESSES en LOCATIONS en dat tussen LACATIONS en SUBJECT DATA BASES. Daaruit volgen natuurlijk geen gegevens over het verkeer tussen de locations. Een matrix die het verband aangeeft tussen lokaties en transakties doet dat wel omdat via Transaktie analyse per transakties de hoeveelheid verkeer is gegeven.
Per vestiging wordt dus vastgelegd welke transakties daar worden uitgevoerd en welke gegevensverzamelingen worden gebruikt. Uiteraard gaat het nu alleen om transakties die gedistribueerde gegevens gebruiken. Natuurlijk kan worden aangegeven hoe de gegevensverzameling wordt gebruikt, maar dat is beschreven op het transaktieschema en niet van belang voor het verkeer. Wil men ook het verkeer binnen een vestiging tussen computersystemen in kaart brengen dan kan dat door de matrix te splitsen in delen per afdeling. Het LAN moet de gegevens binnen de vestiging transporteren van het ene systeem naar het andere. Of de koppeling tussen de systemen mogelijk is, hangt van een aantal faktoren af, maar de hoeveelheid verkeer en de wederzijdse belasting kunnen nu aangegeven worden. In Fig. 54.11 is het geheel in beeld gebracht.
Ook nu blijkt weer dat Transaktie-ontwerp niets in systeemontwikkelingsmethoden overbodig maakt, maar een aanvulling vormt die altijd ingepast kan worden.

- Kies een vorm van gegevensdistributie en bepaal wanneer het gaat om synchrone en wanneer om a-synchrone databases.
- Kies een of meer mogelijkheden voor de lokatie voor gegevensopslag.
- Bepaal per vestiging welke transakties nu te maken krijgen met direkt en indirekt verkeer. Als dit resulteert in interaktief (batch-)verkeer, omdat het gaat om synchrone databases, maak dan voor die transakties decentrale transaktieschema's. Herhaal dit per mogelijkheid voor de lokatie van gegevensopslag.
- Per mogelijheid voor de lokatie van gegevensopslag is nu een set centrale en decentrale transaktieschema's beschikbaar. Voer nu voor alle transakties Transaktie analyse uit, voorzover dat nog niet gebeurt is. Als tijdens het logisch ontwerp de logische Transaktie analyse is uitgevoerd op basis van zijn Transaktieschema's (zie Fig. 54.7) dan gaat het nu dus alleen om wat aanpassingen van die detailschema's
- Maak per mogelijkheid voor de lokatie van gegevensopslag de

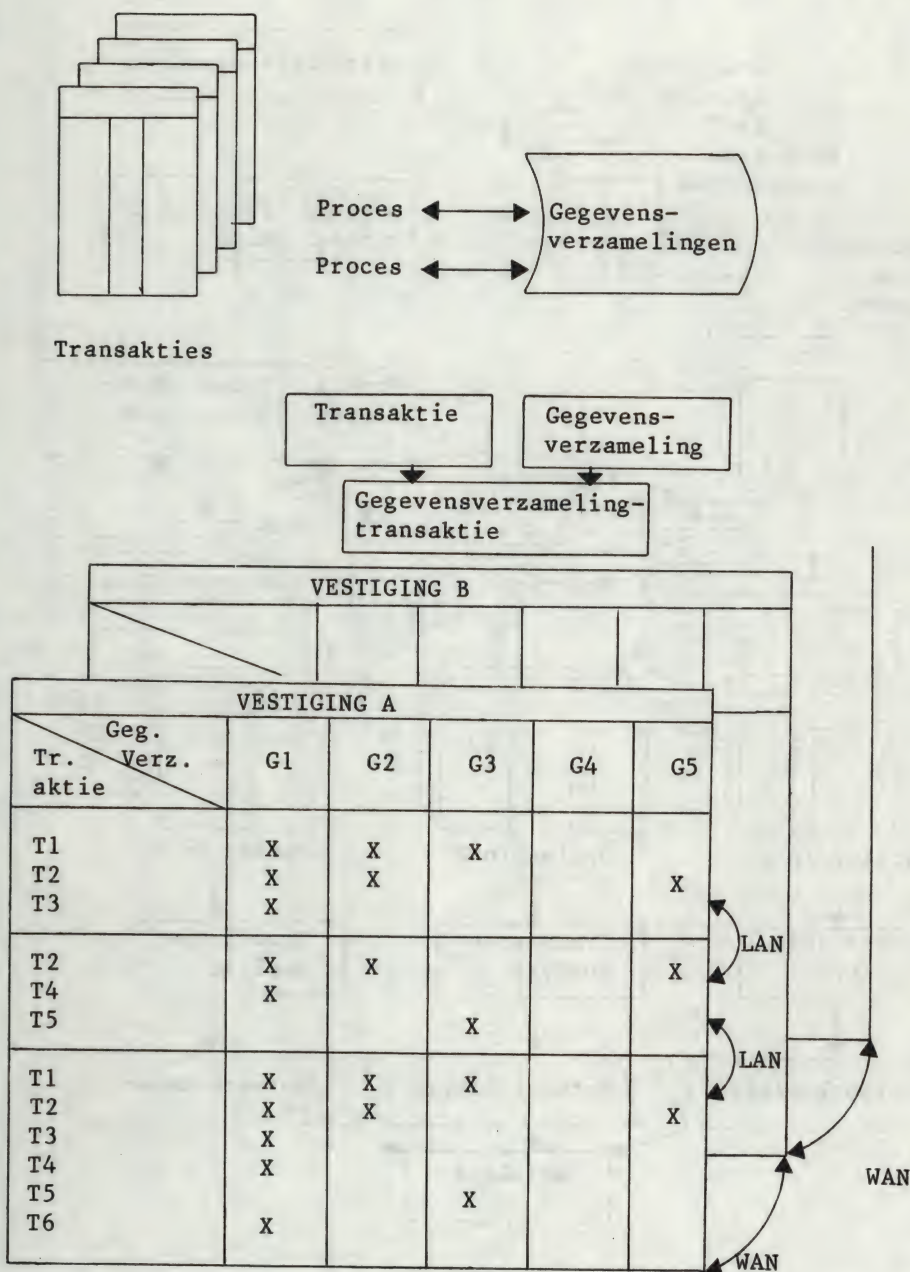


Fig. 54.12 De geografie van transakties en gegevensverzamelingen.

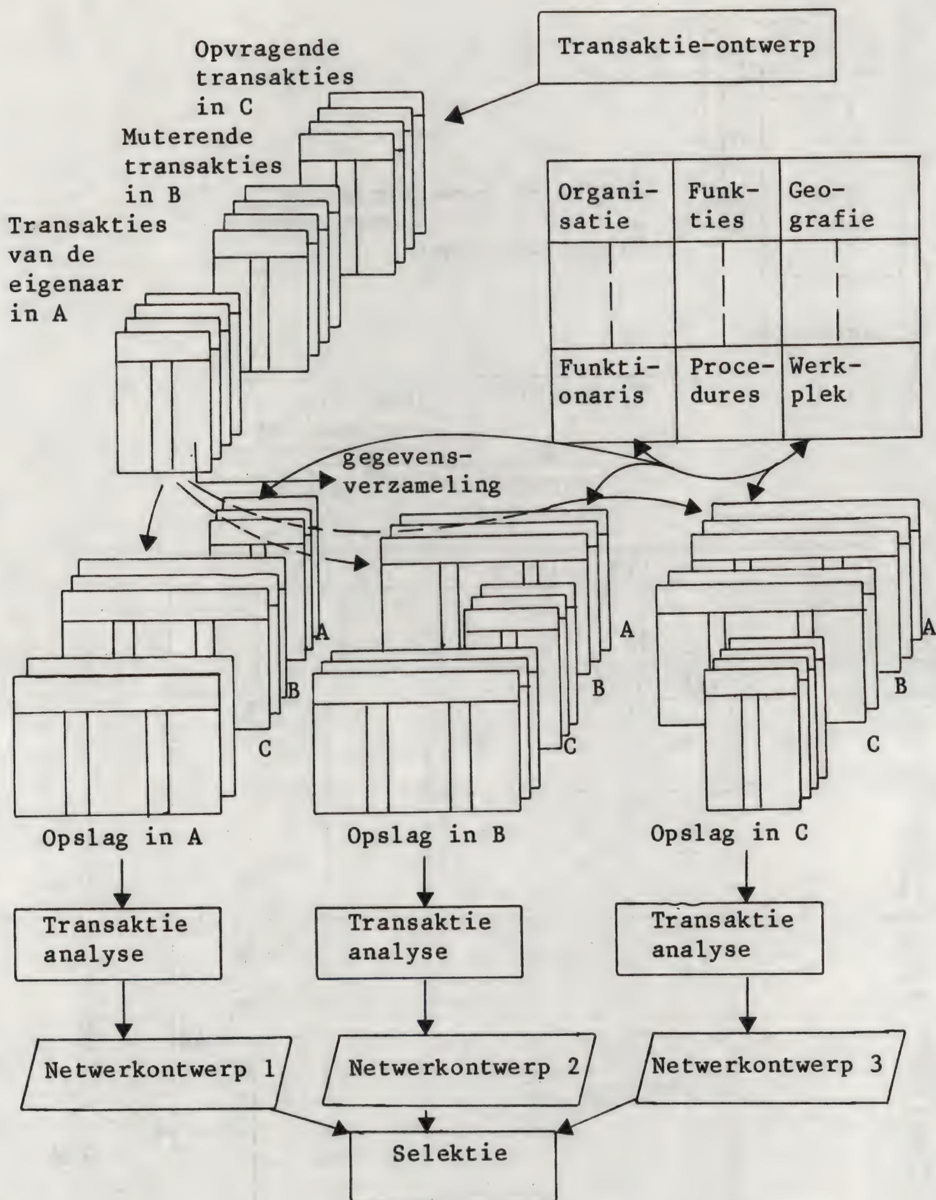


Fig. 54.13 Netwerkontwerp in een distributieve omgeving.

topologie van een netwerk.

- Bepaal op grond van de verkeersresultaten van Transaktie analyse de hoeveelheid verkeer per lijn. Zie de paragraaf Transaktie analyse in distributieve omgevingen en Netwerkontwerp en verkeer. Fig. 54.3, 54.12 en 54.13 brengen het geheel in beeld.
- Overweeg per netwerk wat de mogelijkheden zijn voor alternatieven zoals is aangegeven in de paragraaf Netwerkontwerp en verkeer.
- Uit de nu ontworpen netwerken zal er een moeten worden gekozen op grond van zaken als:
  - Kosten
  - Responsetijden
  - Systeembelasting
  - Konfiguraties
  - Mogelijkheden voor koppeling van systemen.

  We zullen deze punten kort bespreken.
- Kosten.

In eerste instantie gaat het bij de vergelijking tussen de verschillende netwerken om de kosten van apparatuur en lijnen. Zoals is aangegeven in de paragraaf Netwerkontwerp en verkeer kunnen op grond van het netwerkkoncept een aantal vormen gekozen worden om het netwerk te realiseren zonder dat er funktioneel iets verandert. Als men van een sternetwerk overgaat op een boomstruktuur heeft dat gevolgen voor de kosten maar niet voor de funktie. In distributieve omgevingen komen soms maasvormige netwerken voor die niets anders zijn dan een verzameling point-to-point-verbindingen. Men kan bewust een maasvormig netwerk ontwerpen om bijvoorbeeld te beschikken over alternatieve routes. Dan spelen er dus meer aspekten een rol dan alleen verkeer ten dienste van toepassingen. Zo kunnen de uiteindelijke netwerkkosten aanmerkelijk hoger worden wanneer gebruikers duidelijk eisen stellen aan beveiliging, beschikbaarheid of uitwijk. Als echter de gebruiker eisen stelt zullen netwerkleveranciers nauwkeurige aanbiedingen kunnen maken. Dat soort aspekten van het netwerkontwerp heeft niets te maken met het verkeer tengevolge van interaktieve toepassingen. Dat verkeer bepaalt het netwerk dat nodig is om de toepassingen goed te laten werken. Andere eisen betekenen toevoegingen of aanpassingen van dat netwerk.

- Responsetijden.

Alle responsetijden die verkeer met een ander computersysteem inhouden worden kritisch bekeken. De responsetijd wordt nu bepaald door lokale verwerking + transportvertraging van het netwerk + verwerking op een ander systeem. De verwerking op het lokale systeem is in kaart gebracht op het detailschema, de transportvertraging is bekend nu het netwerk is ontworpen, de verwerking op het andere systeem is eveneens aangegeven op het detailschema. De

samenstellende elementen van de responsetijden worden besproken in een team bestaand uit een informatie-analist, een systeemontwerper, een systeemspecialist, een transaktie-analist, een datakommunikatiespecialist en de projektleider. Dat lijkt een duur gezelschap, maar problemen met responsetijden achteraf oplossen is veel duurder dan er tijdens het ontwerp met de juiste mensen naar kijken.

- Systeembelasting.

Beeldschermen die via een lokaal systeem gebruik maken van gegevens op een ander systeem vormen een belasting voor dat systeem. Evenals bij de systeembelasting van de lokale computer gaat het ook hier om het aantal ENTER's per tijdseenheid en het aantal I/O's per tijdseenheod. Voor het andere systeem is ieder bericht heen van de lokale computer een ENTER. Als een lokale transaktie met een T.T.T. van 20 sekonden per transaktie een keer gegevens opvraagt bij een mainframe, zijn dat per uur 180 ENTER's. Als per gegevensuitwisseling 80 response-eenheden op het mainframe moeten worden uitgevoerd, zijn dat 4 response-eenheden per sekonde. Als de transaktie op 10 beeldschermen wordt uitgevoerd, betekent dat voor het mainframe 1800 ENTER's per uur en 40 response-eenheden per sekonde.

- Konfiguraties.

Bij de bepaling van een konfiguratie gaat het in dit verband om twee zaken: de systeembelasting door interaktieve toepassingen en de benodigde schijfkapaciteit voor de distributieve opslag van gegevens. De belasting door interaktieve toepassing van zowel lokale computers, als van de computers waarmee gekommuniceerd wordt kan nu bepaald worden, zie de paragraaf Resultaten en konklusies. (53.1) De benodigde schijfkapaciteit voor het distributieve gebeuren hangt af van de vorm van distributie waarvoor men kiest. Het maakt voor een micro in het algemeen wel wat uit of men kiest voor replicated data of voor subset data. Tijdens het systeemontwerp moet een vorm van distributie gekozen worden en de daaruit voortvloeiende hoeveelheid gegevens die moet worden opgeslagen moet worden bepaald.

Tenslotte kan de konfiguratie nog beinvloed worden door het netwerkontwerp. Als de keuze op Datanet 1 is gevallen, heeft dat gevolgen voor de aan te sluiten systemen. Als men kiest voor een multidrop-netwerk, moeten er wel mini's of micro's gezocht worden die als tributary station kunnen fungeren.

- Mogelijkheden voor koppeling.

In het algemeen zullen distributieve systemen gebouwd worden in omgevingen waar de automatisering lang geleden is begonnen. Als alles nieuw wordt aangeschaft, moet er ernstig worden overwogen alle systemen van een leverancier te nemen of van leveranciers waarvan bekend is dat ze elkaars standaards hebben overgenomen.

In de praktijk komt dat weinig voor. Vaak wordt een mainframe gekoppeld aan decentrale mini's of micro's. De evaluatie van de mogelijkheden voor de koppeling kan het best gebeuren aan de hand van de lagen van het OSI-model van ISO. Per laag wordt onderzocht of de systemen over funkties beschikken om te kommuniceren. Op die manier wordt bepaald wat er in de applikatiesoftware nog moet gebeuren of welke protocol-convertors nodig zijn. De veiligste weg is om van alle benodigde funkties aan de leveranciers situaties te vragen waarin de koppeling in bedrijf is. Als men de gelegenheid krijgt op een dergelijke site een kijkje te nemen, is het van belang ook dan per laag van OSI-model te evalueren hoe het nu precies allemaal is gegaan na de installatie van de hardware.

54.6 Netwerkontwerp en uitwijk

Naarmate de computer voor steeds meer bedrijven een onmisbaar onderdeel vormt van het bedrijfsgebeuren, gaan steeds meer direkties zich afvragen wat een kalamiteit voor gevolgen kan hebben. We beperken ons nu tot die vormen van computeruitwijk die te maken hebben met het netwerk. Het technische aspekt van de computeruitwijk moet deel uitmaken van het bedrijfsrampenplan. Bij een ramp moet er wel meer geregeld worden dan alleen de computeruitwijk, die bovendien meer omvat dan alleen het omschakelen van gebruikerswerkplekken van het bedrijfssysteem naar het systeem van het uitwijkcentrum. Uitgangspunten bij het ontwerpen van een netwerk dat geschikt is voor uitwijk, is gebaseerd op de situatie dat de beeldschermen op de werkplekken in takt zijn. De computerruimte inclusief control units en FEP's is onbruikbaar geworden. In Fig. 54.13 is een voorbeeld getekend. In grote lijnen komt het netwerkontwerp neer op het opstellen van remote control units of FEP's op een plaats die ver genoeg verwijderd is van de computerruimte om te allen tijde ombeschadigd te blijven. In de normale situatie zijn de beeldschermen van de gebruikers via hun clustercontrollers aangesloten op een remote control unit of FEP in het schakelcentrum. Deze control unit of FEP is door een d.c.-verbinding gekoppeld aan een lokale control unit of FEP in de computerruimte. In geval van een kalimiteit wordt de remote control unit of FEP via het uitwijknetwerk gekoppeld aan een control unit of FEP in het computeruitwijkcentrum. In het algemeen zal de afstand tussen het schakelcentrum en de computerruimte het gebruik van goedkope lokale lijnen van een hoge snelheid mogelijk maken. Het huren van eenzelfde lijn naar het uitwijkcentrum kan kostbaar worden. Maar als tijdens het transaktie-ontwerp meteen is vastgesteld welke transakties ook in een uitwijksituatie uitgevoerd moeten kunnen worden, zijn de verkeersattributen de basis voor

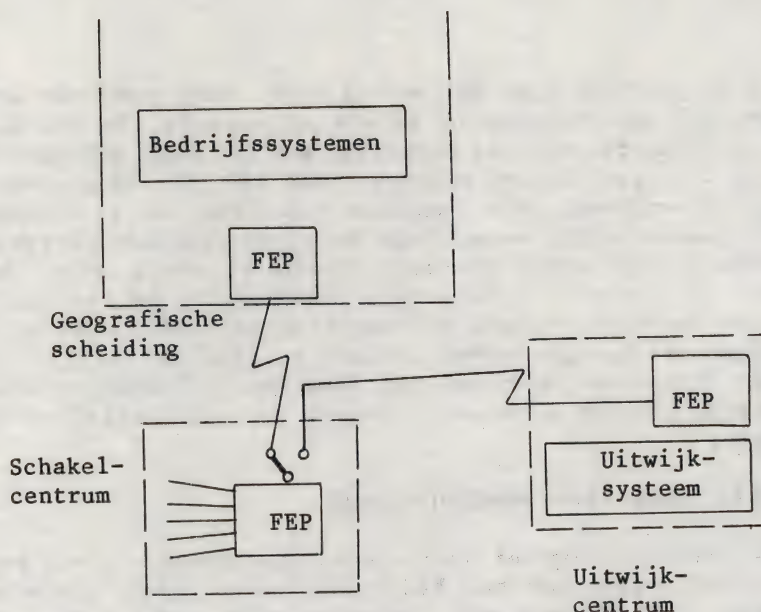


Fig. 54.14 Voorbeeld van computeruitwijk.

het uitwijknetwerk. Dan kan blijken dat er in de uitwijksituatie met een lagere lijnsnelheid kan worden gewerkt en er kan worden uitgerekend welke degradatie van de responsetijden dat betekent.
Naast het ontwerpen van het netwerk is er soms nog een probleem rond het bepalen van de systeembelasting. Als twee operationele rekencentra als elkaars uitwijkcentrum moet kunnen fungeren, is de vraag hoe ze belast worden tijdens de uitwijk. Aan beide kanten moet worden bepaald welke transakties kritisch zijn en wat alle kritische transakties samen in de uitwijksituatie voor de systemen betekenen. De systeembelastingsattributen geven inzicht in de zwaarte van transakties. Daarmee is aangegeven dat computeruitwijk vanaf het begin in het ontwerp kan worden betrokken. In reeds bestaande situaties moeten deze aspekten achteraf geanalyseerd en in kaart worden gebracht.
De hele aanpak om te komen tot een allesdekkend koncept van computeruitwijk is samengevat in de Projektaanpak Uitwijk (PAUW). (24)

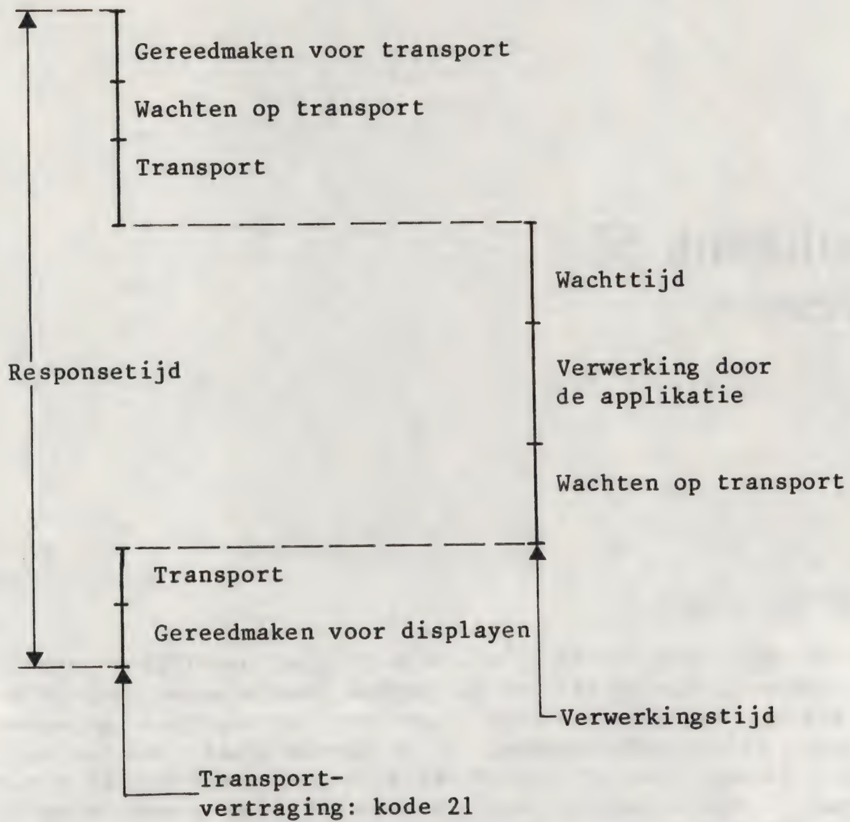
Hoofdstuk 55
Diversen

55.1 Responsetijden

Responsetijden zijn in de praktijk het meest ongrijpbare aspekt in het ontwerp. Tevens zijn ze de oorzaak van de meest voorkomende irritatie. Bij klachten over responsetijden moeten we minstens twee zaken in de gaten houden. In de eerste plaats kan het zijn dat de irritatie van de gebruikers door heel iets anders wordt veroorzaakt, wat ze echter moeilijk onder woorden kunnen brengen. Dat kan een verzameling oneffenheden in de dialoog zijn. Vaak zaken waarvan automatiseerders zich afvragen waarom ze dat nu nog niet weten. Het staat toch allemaal duidelijk in de gebruikershandleiding! In de tweede plaats kan de irritatie veroorzaakt worden door zaken die buiten het werken met beeldschermen liggen. Het kan de tegenzin tegen automatisering in het algemeen zijn. Het kan ook zijn dat men vindt dat men als gebruiker niet gekend is in het ontwerp. Bij klachten over responsetijden kan het nooit kwaad eens op de werkplek te gaan kijken.

- Samenstellende elementen van responsetijden, zie Fig. 55.1.

Alvorens ons te verdiepen in de aanpak van het probleem zullen we eerst de samenstelling van de responsetijd nader bezien. Een responsetijd bestaat uit twee hoofddelen: de transporttijd en de verwerkingstijd. De transporttijd is de tijd die nodig is om het bericht te transporteren. Die valt duidelijk in twee delen uiteen: de tijd voor het transport naar de computer en de tijd voor



$$\text{Verwerkingstijd bij dialoogresponses} = \frac{6 \times 22}{5}$$

$$\text{Verwerkingstijd bij afsluitresponses} = \frac{8 \times 22}{7}$$

Fig. 55.1 Samenstelling van de responsetijden

het transport naar het beeldscherm. We spreken van transporttijd heen en transporttijd terug. Het transport kan betekenen: het verzenden van een bericht over een kabel van een paar meter, over een privenetwerk, over een openbaar pakketgeschakeld net, via een satelietverbinding of een kombinatie van deze mogelijkheden. Naast het transport zelf kunnen er ook nog op allerlei punten wachttijden ontstaan, die toenemen naarmate het net zwaarder wordt belast.
De transporttijd heen bestaat maximaal uit drie onderdelen: gereedmaken voor transport, wachttijd en het transport zelf. Het gereedmaken voor het transport kost meestak weinig tijd bij een beeldscherm. Het gaat om het selekteren van de te verzenden velden, het toevoegen van besturingstekens en dergelijke. Bij beeldschermen speelt dit meestal geen rol. Soms moet er bij een computer-computerverbinding een heel stuk verwerking plaatsvinden. Bij een micro-mainframeverbinding kan dat zelfs wel even duren!
Het wachten voor de lijn komt alleen voor, wanneer verscheidene beeldschermen gebruik maken van dezelfde lijn. Dat kan een multipointlijn zijn of een point-to-pointlijn met een clustercontroller. Bij een multipointlijn hangt de lengte van de pollcyclus af van de drukte, bij een clustercontroller van de ingestelde pollfrekwentie. Bij asynchrone beeldschermen die zijn aangesloten via een multiplexer speelt de wachttijd weer geen rol, wanneer het gaat om eenvoudige multiplexers. Bij pogingen om nog meer beeldschermen op een lijn aan te sluiten kan van zeer intelligente multiplexers gebruik worden gemaakt, die het transport uitsmeren over de beschikbare kapaciteit. In dat geval kunnen dus berichten een korte tijd worden vastgehouden in de buffers van de multiplexers. Dan treden dus weer wachttijden op. Die wachttijden hangen weer af van drukte. Heel goed is dat merkbaar, wanneer interaktief verkeer wordt gekombineerd met batch-verkeer. De batch vormt een te zware belasting voor de lijn tussen de multiplexers en dat merken de beeldschermgebruikers.
Het derde element in de transporttijd heen is uiteraard het transport zelf. Dit is de tijd die nodig is om een bericht van een bepaald aantal tekens te versturen over een lijn met een bepaalde kapaciteit. Het duurt precies een seconde om een bericht van 300 tekens te versturen over een lijn met een kapaciteit van 300 tekens per seconde. Zo eenvoudig als dit lijkt zo moeilijk blijkt het voor ontwerpers de berichtlengte vast te stellen. De gemiddelde programmeur/ontwerper schrikt zich een ongeluk wanneer hij een dump ziet van het aantal tekens dat bij een bepaalde interaktie over de lijn gaat. Bij domme, teletype-achtige beeldschermen gaat bij iedere aangeraakte toets een teken over de lijn. Bij 3270-achtige beeldschermen gaat een aantal velden inclusief allerlei besturingstekens over de lijn. Daarnaast is het

natuurlijk zo, dat blanks niet op het scherm te zien zijn, maar wel over de lijn getransporteerd worden. In bestanden en databases wordt vaak met velden van vaste lengte gewerkt. Een veldlengte voor een bedrijfsnaam van 35 tekens is niets bijzonders en voor straat en woonplaats vaak hetzelfde. In de records worden de velden eenvoudig aangevuld met blanks. Wanneer door de applikatie die velden getoond worden op het scherm, dan worden ze naar het beeldscherm gestuurd. In toepassingen waar een lijst van N.A.W.-gegevens op het scherm gezet moet worden is de hoeveelheid blanks niet meer verwaarloosbaar. Bij openbare datanetten, waar per teken betaald moet worden, is er dus nog een andere reden om nog eens kritisch te kijken naar het aantal te transporteren tekens. Kortom, het gaat bij dit element om de applikatie en om de intelligentie van het beeldscherm. Helaas blijken veel ontwerpers en programmeurs daar weinig van te weten. Met de toepassing van TP-monitoren en vierde-generatietalen neemt de "afstand" tot de toegepaste beeldschermen duidelijk toe. Een verontschuldiging kan dat nooit zijn, gezien de omvang van de problemen die opgelost moeten worden wanneer responsetijden te lang zijn.
Het tweede deel van de transporttijd is de tijd die nodig is voor het terugzenden van het bericht. Ook hier kan een groot verschil bestaan tussen de gedisplayde tekst en de te transporteren tekens. Met name bij domme, teletype-achtige beeldschermen kunnen gemakkelijk 3 tot 4 maal zoveel tekens over de lijn gaan als er op het scherm staan. Ook hier geldt dus weer, dat ontwerpers die iets aan responsetijden willen doen, dit soort zaken moeten uitzoeken. In de praktijk gaat het echter in een bedrijf vaak maar om een computersysteem, dus hoeft het maar één keer te worden uitgezocht. Daarna kan die kennis meegenomen worden in Transaktie analyse en worden verwerkt in detailschema's.
De verwerkingstijd is de tijd die de computer nodig heeft om het te ontvangen bericht te verwerken en het bericht terug te formeren. Dit deel valt in 3 onderdelen uiteen: de wachttijd voordat de applikatie gestart is, de verwerking door de applikatie en de wachttijd voor het bericht de lijn op gaat. De eerste wachttijd wordt voornamelijk bepaald door een wachtrij die ontstaat, wanneer het systeem overbelast begint te raken en door de tijd die het operating systeem nodig heeft om een applikatie te starten. Dat laatste wordt weer bepaald door de hoeveelheid beschikbare ruimte die vrij is in het geheugen, het aantal schijven en de drukte op het systeem. De verwerkingsstijd wordt bepaald door de applikatie. In de praktijk is het knelpunt het aantal I/O-operaties en de tijd per I/O-operatie. Op de tijd per I/O-operatie komen we later in deze paragraaf terug. Voor wat betreft het aantal I/O-operaties kunnen tijdens het logisch ontwerp al fouten worden gemaakt. Vooral het werken met te veel hulp- of stuurbestanden

kan fnuikend zijn voor de responsetijden.Aangezien deze zaken meestal tot de architektuur van het systeem behoren, is het vaak zo, dat daar achteraf niets meer aan te veranderen is. Daarom is het van het begin af aan noodzakelijk konkreet vast te stellen, wat in de diverse situaties de eisen zijn ten aanzien van de responsetijden (dialoogsimulatie) en te begroten hoe de zullen worden (Transaktie analyse).
Het laatste deel van de verwerkingstijd wordt veroorzaakt door de wachttijd voor het transport op de lijn. Die tijd zal afhangen van het aantal beeldschermen aan de lijn, de kapaciteit van de lijn en de bezetting.
In Fig. 55.1 is het geheel in beeld gebracht. De cijfers in de figuur slaan op de parameterkodes van Transaktie analyse. De formules geven de manier van berekening door het rekenprogramma aan. In dit verband zij er nogmaals op gewezen dat met de parameterkodes 6 en 8 de verwerking door de applikatie in kaart wordt gebracht. De andere elementen van de verwerking worden in de vorm van een gemiddelde aanname ondergebracht in parameterkode 21. In de praktijk blijkt 0,1 sec. een goede startwaarde te zijn.
- Meetbare en beinvloedbare elementen.
Laten we vervolgens eens bekijken welke elementen van de responsetijden beinvloedbaar of meetbaar zijn. Beinvloeding is van belang bij te ontwerpen systemen, meting bij de evaluatie van bestaande systemen. Transporttijden zijn bij toepassing van Transaktie analyse in een vroegtijdig stadium bekend. Immers, Transaktie analyse levert gegevens over aantal beeldschermen per lokatie, bezetting van die beeldschermen en het verkeer. In die situatie is het netwerk dus aan te passen aan de benodigde kapaciteit. In een bestaande situatie moet onderzocht worden waar de knelpunten zitten. Men kan nu achteraf Transaktie analyse uitvoeren om achter de hoeveelheid verkeer en de lijnbezetting te komen. Het is natuurlijk ook mogelijk om het verkeer te analyseren met een meetapparaat. Wanneer dit apparaat goed is te programmeren, dan kunnen er dezelfde cijfers uitkomen als uit Transaktie analyse: gemiddelde berichtlengte heen, gemiddelde lengte terug en de tijd tussen de berichten. Bij een netwerk van enige omvang verdient de aanschaf van een dergelijk apparaat aanbeveling. Een eenvoudige "meekijker" op de lijn heeft maar beperkte mogelijkheden. Het moet een apparaat zijn, waarbij kan worden ingesteld vanaf welk teken of soort teken het loggen van de lijn moet beginnen en tot welk punt in het bericht. Apparaten die alleen een dump maken gedurende een bepaalde tijd zijn in dit verband niet interessant. Een transaktie-analist begrijpt om welke gegevens het hier gaat.
Dan komen we bij de beinvloeding van de tijd die de computer nodig heeft voor de verwerking. Het eerste element is de wachttijd

voordat de applikatie gestart wordt. Bij de beinvloeding gaat het om twee aspekten: de belasting en de software. Er is bij iedere computer een duidelijke relatie tussen belasting en responsetijden. Wat meestal zo onduidelijk is, is de reden voor die (over)-belasting. In paragraaf 33.2 is duidelijk gemaakt dat de verantwoordelijkheid voor responsetijden is teruggebracht bij de leverancier, wanneer een bedrijf zijn werk goed doet en de belasting uitdrukt in cijfers waarmee de computerleverancier kan werken. Beinvloeding tijdens het ontwerp is dus goed mogelijk. Meting achteraf is meestal ingewikkelder, omdat men niet goed weet of het nu aan het operating systeem ligt of aan de applikatie. Sommige computersystemen zijn te voorzien van meetfunkties. Uit de output van die programma's is dan wat af te leiden over I/O-operaties, geheugengebruik en dergelijke. Uiteindelijk is er advies nodig van de leverancier over verbetering door geheugenuitbreiding en vergroting van het aantal schijven. Dan moet garantie gevraagd worden over de verbetering of er wordt een proef gedaan. Wanneer transakties niet met behulp van Transaktie analyse in kaart zijn gebracht en er dus niet zoveel zicht is op de totale belasting, dan bestaat er een duidelijk probleem. Het werk dat tijdens het ontwerp had moeten gebeuren moet alsnog gedaan worden. Maar dat betekent nog niet, dat de oplossing eenvoudig is. Direkt bij de start een grotere computer kiezen op basis van een goede begroting van de belasting via Transaktie analyse, is nu eenmaal wat anders dan achteraf overschakelen op een grotere machine!
Dan komen we bij de software. De computerleverancier bedoelt daarmee de mate waarin goed gebruikt is gemaakt van de geboden funkties. COBOL blijft natuurlijk COBOL, maar iedere computer heeft zijn specifieke eigenschappen. Een toepassing die op de ene machine goed draait, kan vastlopen op een andere. Vooral de I/O-operaties moeten door programmeurs nauwgezet bestudeerd worden. Het is een algemeen voorkomend verschijnsel, dat programmeurs na afloop van hun eerste projekt op een machine het liefst de hele software in de prullenbak zouden willen gooien en opnieuw willen beginnen. Een grondige bestudering in theorie en praktijk van alle geboden systeemfunkties is op z'n plaats. Het is te hopen dat daarbij ook te leren is wat men nu juist niet moet doen.
Een heel ander aspekt is het programmaontwerp. Programma's behoren goed gestruktureerd ontworpen te worden om allerlei redenen. De tijd van houtje-touwtjeprogramma's zou voorbij moeten zijn. Toch is het bij de beste struktuur meestal niet zo, dat de eisen ten aanzien van de responsetijden verwerkt zijn. Bij struktuur gaat het primair om de mooie struktuur en de voordelen ervan. Wie maakt een struktuur onlogisch terwille van de responsetijd? Dat doet niemand. Ten eerste niet omdat er meestal geen exakte eisen

zijn ten tweede omdat "responsetijdproblemen maar anders moeten worden opgelost". Bij de meeste interaktieve systemen is het zo, dat de vereiste responsetijden niet altijd flitsend hoeven te zijn. In Transaktie analyse wordt niet voor niets onderscheid gemaakt tussen dialoogresponses en afsluitresponses. Dat betekend, dat tijdens het logisch ontwerp al moet vastliggen welke eisen per interaktie gelden, zodat daarmee al rekening kan worden gehouden bij het ontwerpen van de programmastruktuur. Bij kritische interakties moeten alle voor het displayen niet strikt noodzakelijke I/O-operaties worden uitgesteld tot de eerstvolgende afsluitresponse. Beinvloeding is dus mogelijk wanneer dialoogsimulatie is uitgevoerd tijdens het logisch ontwerp. Dan worden geen overdreven, maar akseptabele responsetijdeisen vastgesteld. Bij de Transaktie analyse die erop volgt, kan dan al heel vroeg worden vastgesteld of de eisen haalbaar zijn of niet. Meting van dit element van de responsetijd achteraf is vrij eenvoudig, om dat er meestal voldoende dokumentatie beschikbaar is. Voor de kritische responsetijden is gemakkelijk vast te stellen welke I/O-operaties gedaan worden in de module die de interaktie verzorgt. Als dat soort dokumentatie niet beschikbaar is, is er weinig hoop voor verbetering van de responsetijden. Maar zelfs als die dokumentatie er wel is, betekent dat nog niet dat er iets aan de responsetijden te doen is. Als er geen I/O-operaties verplaatst kunnen worden naar andere interakties binnen de transaktie, dan zou de hele struktuur van bestanden en programma's moeten worden aangepakt. In het algemeen begint men daar niet aan. Responsetijdproblemen moeten dan ook niet opgelost, maar voorkomen worden! In dit verband kan dan ook worden gewezen op de toegangspadanalyse gebaseerd op het dialoogontwerp, zoals is aangegeven in het deel voor de informatie-analisten in de paragraaf Transaktie-ontwerp in verschillende omgevingen.
Wanneer achteraf Transaktie analyse wordt toegepast, gebeurt dat meestal omdat de ontwerpers ook geen inzicht meer hebben in het aantal I/O-operaties. Achteraf uitgevoerde Transaktie analyse levert dan een overzicht van de aantallen I/O-operaties in de diverse transakties. Door op die manier de bestaande situatie in kaart te brengen, wordt in ieder geval voorkomen dat het aksent op het verkeerde element in de responsetijd komt te liggen. Immers, elk van de genoemde elementen kan een knelpunt zijn in het geheel. Het zou triest zijn als er kapitalen werden uitgegeven aan snellere modems terwijl het knelpunt in de applikatie zit. Even vervelend is het, wanneer overgeschakeld is op een systeem met meer geheugen en meer schijven maar de responsetijden nauwelijks verbeterd zijn doordat er teveel verkeer over te langzame lijnen gaat.
Samenvattend kunnen we zeggen: voorkomen is beter dan genezen.

Elementen	Van te voren	Achteraf
Gereedmaken	Bij beeldschermen verwaarloosbaar	
Wachten	Lijnbezetting bepalen via Transaktie analyse	Meetapparatuur of Transaktie analyse
Transport	Kennis van beeldschermbesturing Transaktie analyse	Meetapparatuur of Transaktie analyse
Wachten	Keuze van computer systeem op basis van cijfers	Analyse van operating systeem met systeemspecialisten
Applikatie	Dialoogsimulatie, Transaktie analyse en programma-ontwerp dat daarop is gebaseerd	Analyse van I/O-operaties voor kritische interakties of Transaktie analyse
Wachten	Transaktie analyse	Meetapparatuur of Transaktie analyse
Transport	Kennis van beeldschermbesturing en Transaktie analyse	Meetapparatuur of Transaktie analyse

Fig. 55.2 Overzicht van de samenstellende elementen.

Dat is niet nieuw. Het nieuwe is dat er in dit boek methoden worden aangegeven die daartoe bijdragen. We kunnen deze paragraaf in beeld brengen als in Fig. 55.2 is weergegeven.
Tenslotte nog iets over de tijd per I/O-operatie. Wanneer een I/O-operatie bestaat uit een databasecall, dan hangt de tijd die daarvoor nodig is, onder andere af van het aantal fysieke I/O's dat het systeem moet uitvoeren. Aangezien gegevens meestal op schijf staan en verscheidene toepassingen op hetzelfde moment van dezelfde schijf gebruik maken, ontstaan er wachtrijen van I/O aanvragen in het operating system. Deze toepassingen zijn niet alleen de interaktieve, maar ook de batch- en printtoepassingen. De printtoepassingen kunnen daarbij nog gescheiden worden in printwerk op een centrale printer en printwerk op de werkplek. Bij de beoordeling van de performance van een aan te schaffen systeem is het van essentieel belang Transaktie analyse dusdanig uit te voeren, dat er een duidelijk inzicht ontstaat in het gebruik van de printers. Immers, al deze aspekten dragen bij tot een belasting van de computer en daarmee tot de tijd per I/O-operatie. Wanneer als eis gesteld wordt dat de responsetijden goed moeten blijven, ook als er geprint wordt, dan is het van belang een architektuur te kiezen waarbij printwerk zoveel mogelijk losgekoppeld wordt van de interaktieve toepassingen. Bijvoorbeeld door aparte schijven en lijnen naar remote printers. Zoals al eerder opgemerkt, is het zeker binnen een netwerk verstandig interaktief verkeer zoveel mogelijk onafhankelijk van batch-verkeer te maken. Natuurlijk zal van de I/O-operaties ook afhangen van het aantal aktieve terminals. Transaktie analyse biedt de mogelijkheid om met de parameter "Tijd per response-eenheid" te experimenteren en vast te stellen in hoeverre de berekende totale responsetijden gevoelig zijn voor schommelingen in de tijd per I/O-operatie. Op die manier kunnen variaties in de belasting gekwantificeerd worden naar variaties in responsetijden.

- Eisen stellen ten aanzien van responsetijden.

Het heeft geen enkele zin bij de opzet van een nieuw systeem een zin in de specifikaties op te nemen die iets zegt over de vereiste gemiddelde responsetijd. In de eerste plaats omdat het gemiddelde zonder een spreiding een nietszeggend begrip is. De computerleverancier haalt de schouders er over op, omdat hij niet weet wat voor applikaties er gebouwd zullen worden. De ontwerper zegt dat hij zijn best zal doen, maar hij heeft geen idee waar uiteindelijk de responsetijden terechtkomen.
In de tweede plaats bestaat er voor de gebruikers evenmin een gemiddelde responsetijd. Van sommige lange responsetijden is hij zich helemaal niet bewust, andere moeten flitsend zijn. Het maakt nogal uit of hij met een ingewikkeld dossier bezig is en daar een beeldscherm bij gebruikt of dat hij zit te bladeren op een beeld-

scherm. Bovendien: hoe zou de gemiddelde programmeur een gemiddelde responsetijd moeten realiseren? Dat geeft zeker problemen wanneer zes programmeurs twintig applikaties bouwen. Eisen ten aanzien van responsetijden hebben alleen zin, wanneer ze toegespitst zijn en er vakmanschap in huis is om er konkreet aan te werken.
Wanneer tijdens het logisch ontwerp de transaktieschema's gereed gekomen zijn, kan een ervaren informatie-analist op het transaktieschema al aangeven welke responsetijden kritisch zijn. Bij dialoogsimulatie moet op de simulator per interaktie de responsetijd kunnen worden ingesteld. Tijdens de simulatie met de gebruiker dient de informatie-analist te letten op het verloop van de transaktie. Welke interakties zijn gevoelig voor lange responsetijden en welke niet of in mindere mate? Van de kritische wordt de responsetijd steeds met een seconde verlengd. Op die manier is in overleg met de gebruiker goed vast te stellen wat nog akseptabel is. Daarbij is het uiteraard van belang steeds de maximaal akseptabele tijd vast te stellen. Op de schermlay-out, zoals die door de simulator worden geprint, liggen die cijfers nu vast. Ze kunnen ook opgenomen worden in het transaktieschema door het aantal seconden boven de pijl naar links te zetten. Deze eisen moeten vanaf het begin een rol spelen bij de toegangspad-analyse, het programma-ontwerp en het database-ontwerp. Deze moeten al in het begin zijn toegesneden op het halen van die responsetijden. Dan wordt in ieder geval bij de kritische responsetijden het aantal databasecalls en het aantal fysieke database-operaties meteen geminimaliseerd. Strukturele aanpassingen achteraf worden bijna nooit (meer) uitgevoerd. Nog steeds worden er databases ontworpen volgens de regels van de theorieboeken, maar zonder enige visie op performance.
- Het realiseren van eisen ten aanzien van responsetijden.
Er bestaat uiteraard geen eenvoudig regeltje om de responsetijd binnen bepaalde grenzen te houden. In geen enkele systeemontwikkelingsmethode wordt gerept over het ontwerpen van systemen met een bepaalde performance. In de meeste methoden komt men qua konfigureren zelfs niet verder dan wat plaatjes met beeldschermen, printers, overige periferie en "hardware-dozen". Bij de analyse van de elementen waaruit de responsetijd is opgebouwd, hebben we kunnen zien dat het niet alleen gaat om verschillende soorten elementen, maar ook om elementen waarover door heel verschillende deskundigen op heel uiteenlopende momenten wordt beslist. Er ligt bijvoorbeeld meestal nogal wat tijd tussen de keuze van een computersysteem en de bouw van de programma's. Dat er bij beide aktiviteiten meestal ook verschillende mensen zijn betrokken is duidelijk. In deze verscheidenheid van samenstellende elementen en soorten deskundigheid ligt een belangrijke oorzaak van het uit

de hand lopen van responsetijden. Alleen wanneer op alle desbetreffende gebieden de puntjes op de i gezet worden, bestaat er een kans op goede resultaten. De systematiek wordt in de methoden en de manier van werken aangegeven.

- Geen systeem aanschaffen voor het logisch ontwerp is uitgevoerd waarin zijn opgenomen de methoden dialoogsimulatie en Transaktie analyse.
- Tijdens de gesprekken met de computerleverancier gekwantificeerde eisen stellen zoals is aangegeven in paragraaf 33.2.
- Tijdens het technisch ontwerp konsekwent de hoeveelheden I/O's van kritische responsetijden in de gaten houden.
- Tijdens de bouw regelmatig afrekenen met de bouwer over de gerealiseerde toepassingen en de erbij behorende responsetijden.
- Netwerkkoncepten konsekwent doorrekenen op basis van de resultaten van Transaktie analyse.

Voor reeds bestaande systemen hetzij groot of klein is geen enkele hoop, als nooit systematisch de transakties in kaart zijn gebracht. Als de responsetijden slecht zijn, zullen ze dat wel blijven. Soms worden er door het topmanagement ingrijpende beslissingen genomen om van de altijd klagende gebruikers af te komen: afbouw van het centrale computersysteem en iedere vestiging of groep gebruikers een eigen computer met daarop dezelfde toepassingen. Het lijkt een aardige oplossing. Per computer minder beeldschermen en dus minder belasting, dus een kleinere computer. Maar als de toepassingen nu weer niet in kaart gebracht worden wat de belasting betreft, dan heeft binnen korte tijd iedere vestiging weer hetzelfde probleem, maar nu op z'n eigen minicomputer. Dan slaat decentralisatie wel op decentralisatie van de problemen. Of het bedrijf er in z'n totaliteit op is vooruitgegaan is natuurlijk de vraag.

Wanneer tijdens het ontwerp de responsetijden de aandacht krijgen als boven aangegeven, dan kan het zijn dat dit aanleiding geeft tot kostbare hardware. Als we dat hebben vastgesteld is er een zeer belangrijk punt bereikt. Enerzijds omdat dat punt normaliter niet vaak bereikt wordt, anderzijds omdat nu zonodig het iteratieve aspekt van het ontwerp in verband met de responsetijden naar voren komt. Gebruikers worden nu gekonfronteerd met de konsekwenties van hun eisen. Zoals bij alle andere bedrijfsinvesteringen begint nu het spel van het afwegen van de eisen tegen de kosten. Wanneer de kosten van de hardware (nog) een rol spelen zullen de gebruikers water bij de wijn van hun eisen moeten doen. Desnoods kan gemakkelijk de dialoogsimulatie worden gebruikt, maar nu met langere responsetijden om duidelijk te maken hoe het systeem zal funktioneren wanneer bijvoorbeeld op een of meer lokaties geprint wordt. Wanneer Transaktie analyse goed is uitgevoerd is natuurlijk ook bekend hoeveel uur per dag deze langere

responsetijden optreden. Ook kunnen op dit moment alternatieven voor het dialoogontwerp worden overwogen. Met name bij zeer kritische responsetijden en vele interakties met het systeem kan vaak worden overwogen verschillende interakties samen te voegen tot een interaktie. Dat betekent tevens minder overhead op het systeem. Nogmaals, het belangrijkste aspekt is dat er nu in een vroegtijdig stadium konkreet met de gebruiker kan worden gepraat over responsetijden op basis van gegevens en alternatieven. Dat is in alle systeemontwikkelingsmethoden een ontbrekend aspekt. Wanneer er in een bedrijf geen tijd is voor dit soort aspekten, dan verdient men ook niet anders dan vele problemen met gebruikers, waarvan de responsetijdproblemen zeker niet de enige zullen zijn. Bij de afwegingen van eisen tegen kosten van alternatieven gaat het om zaken als
- netwerkkoncepten
- mate van decentralisatie
- benodigde netwerkkapaciteit
- tarieven van netwerkhardware of -diensten.

55.2 Evaluatie van interaktieve toepassingen

Hoewel in de praktijk niet zoveel terecht komt van de evaluatie van de gebouwde systemen, gebeurt het soms dat gebruikers een opgeleverd systeem niet aksepteren of er niet mee willen werken. Dan wordt er van hoger hand wel eens opdracht gegeven uit te zoeken wat er nu eigenlijk aan de hand is: klagen de gebruikers ten onrechte of hebben de automatiseerders inderdaad een slecht systeem gebouwd.
Bij de beoordeling van een systeem kan men letten op het koncept, de ergonomie en de performance. Bij de beoordeling van het koncept gaat het bijvoorbeeld om het transaktie-ontwerp. Hoe passen de transakties in de rest van de procedures? Hoe verhoudt de terminaltransaktietijd zich tot de gewenste of noodzakelijke doorlooptijd? Hoeveel moet er geprint worden, waar staan de printers, hoe lang staat men te wachten? Om een beeld te krijgen van de ergonomische aspekten van het systeem dient men op de hoogte te zijn van het kennis- en verwachtingspatroon van de gebruikers.
Dan kan men vervolgens beoordelen
- de schermlay-outs, terminologie en aantal schermen per transaktie
- de samenhang tussen transakties: hoe gemakkelijk kan men van de ene transaktie overschakelen op de andere
- het gebruik van menuschermen, kommando's en funktietoetsen
- foutmeldingen

Bij performance gaat het om de responsetijden. Zoals in de vorige paragraaf is beschreven, bestaat de responsetijd uit een aantal

elementen.
Van elk van die elementen dient men drie aspekten te onderzoeken:
- Ontwerpfouten. Bij databases gaat het vaak om ontwerpen die theoretisch goed zijn, maar die een goede performance in de weg staan. Procesgegevens horen niet in een database thuis, tenzij de responsetijden daar een faktor 10 beter van worden. Bij programmastrukturen worden I/O's op het meest logische moment uitgevoerd, tenzij blijkt dat kritische responsetijden verbeterd kunnen worden door een deel van de I/O's te verplaatsen naar minder kritische programma-onderdelen van afsluitresponses. Het verkeer over de lijn is meestal een gevolg van een eenmaal gekozen of vastliggende dialoogstruktuur tenzij blijkt dat bijvoorbeeld het per ENTER oversturen van een kompleet scherm nadelig is voor de responsetijden.
- Gebrekkige optimalisatie. De bovengenoemde drie elementen kunnen ook verbeterd worden door optimalisatie. Voor de databases is dat het tunen van de database, voor de programma's komt dat neer op het opvolgen van de adviezen van de leveranciers ten aanzien van grootte van programma's en buffers, aantallen buffers en dergelijke. Om het verkeer over de lijnen te optimaliseren is het verstandig een dump te maken van wat er aan tekens over een lijn gaat. De systeemontwerper moet een duidelijk verband kunnen vinden tussen de dialoog aan het beeldscherm en de tekens over de lijn. Een aantal van die tekens heeft te maken met het lijnprotocol en is niet te veranderen. Een deel van de tekens zorgt voor de schermbesturing en ligt ook vast. Heel vaak is het aantal blanks van de velden erg hoog. Er kan onderzocht worden of die te veranderen is door de applikatie aan te passen of door andere funkties van het I/O-systeem te gebruiken.
Het maken van een dump van wat er over de lijn gaat is bij de meeste computersystemen een standaard funktie van het operating system. Zo niet, dan moet met een extern meetapparaat (line spy) het verkeer gelogged worden op een cassette en daarna worden geprint.
- Systeembelasting. Bij systeembelasting gaat het om het aantal I/O's per seconde dat een systeem moet verwerken, gegeven het aantal beeldschermen, transakties en programma's. Het aantal beeldschermen bepaalt het aantal ENTER's. De wait- en think-times bepalen het aantal ENTER's per seconde. De programma's bepalen het aantal I/O's per ENTER. Als tijdens het ontwerp Transaktie analyse .niet is toegepast, kan niemand deze gewenste cijfers produceren. Transaktie analyse wordt nu achteraf uitgevoerd om de bestaande situatie kwantitatief in kaart te brengen. We zullen de aanpak puntsgewijs bespreken.
- Bepaal de relevante transakties per beeldscherm. Welke transakties worden het meest uitgevoerd en over welke transakties klagen

	Kode	Kansfaktor	E(x)	VAR(x)
*** S39 Sel.lev.cond groep***				
Lezen brondokument	12	45	2	5
Tik gegevens	1	45	1	0
Transport op lijn	2	45	1	10
Aantal enters	5	45	1	0
Aantal fysieke I/O's	6	45	2	1
Laden programmadeel	6	45	3	2
Terugzenden scherm S29	3	45	500	0
Interpreteren scherm	12	45	2	2
*** S29 Lev.cond.hand.lev. ***				
Lezen brondokument	12	45	5	10
Tik gegevens in	1	45	3	10
Transport op lijn	2	45	3	10
Aantal enters	5	45	1	0
Aantal fysieke I/O's	6	45	2	1
Laden programmadeel	6	45	3	2
Terugzenden scherm S21	3	45	500	0
Einde transaktie	99			

Fig. 55.3 Detailschema met standaardregels.

de gebruikers. Daarbij kan het best zijn dat ze klagen over een bepaalde transaktie, maar dat de klachten veroorzaakt worden door transakties op andere beeldschermen.

- Transaktieschema's hebben niet de funktie van gebruikersdokument zoals tijden het logisch ontwerp. Ze zijn nu alleen nodig als start voor Transaktie analyse. Als de transaktie-analist de transakties al kent, kan hij zelfs de transaktieschema's overslaan en direkt de detailschema's maken bijvoorbeeld met behulp van een paar standaard parameterregels per ENTER. Zie Fig. 55.3. Neem voor parameter 22: 0.1 sec.
- Maak een tijdbestedingsdiagram aan de hand van de resultaten van Transaktie analyse. Kontroleer met de gebruikers of het berekende aantal beeldschermuren klopt met de werkelijkheid. Zo niet, dan is er iets aan de hand met de cijfers van de gebruikers of het detailschema klopt niet met de werkelijkheid. Kontroleer: typesnelheid, aantal in te typen tekens, denk- en wachttijden en kansfaktoren. Pas als het bezettingsoverzicht klopt met de realiteit zijn de andere resultaten van Transktie analyse te vertrouwen.
- Kontroleer of er transakties zijn waarvan volgens de resultaten de responsetijden veel te hoog zijn. Konfronteer de ontwerpers met het aantal I/O's en bepaal de oorzaak. Als de responsetijden inherent zijn aan de gekozen manier van verwerken hoeven we niet verder te zoeken. Dan hebben de ontwerpers nu een probleem. Dat

kan betekenen een aanpassing van het database-ontwerp en waarschijnlijk van een aantal programma's. Voor alle zekerheid kan parameter 22 bijvoorbeeld gehanteerd worden om vast te stellen of dan de responsetijden akseptabel worden. Als dat het geval is gaat het meestal niet om ontwerpfouten maar om een systeembelastingsprobleem.
- Bepaal een aantal situaties en bereken het aantal I/O's per seconde zoals is aangegeven in paragraaf 53.8.
- In CxN-omgevingen kan de transportvertraging nog een rol spelen. Bepaal aan de hand van de berichtlengte heen en -terug en de lijnsnelheid de transportvertraging. Soms wordt de vertraging voor het grootste deel bepaald door het aantal POLL's per tijdseenheid. Bij sommige systemen kan het datakommunikatiesysteem overbelast raken wat er toe leidt dat de POLL-frekwntie erg laag wordt. In die gevallen is het nuttig een dump te maken van wat er over de lijn gaat. Vergelijk de berichtlengtes met de overeenkomstige resultaten van Transaktie analyse. De dumps zijn altijd voorzien van tijdsaanduidingen. Probeer vast te stellen op welk moment een bepaald beeldscherm wordt gepolled, door in de getransporteerde tekens naar het beeldscherm een terminal-adres te vinden. Bepaal het moment waarop een blok naar het systeem getransporteerd wordt een het moment waarop er een blok terug komt. Als de transaktie enige keren wordt uitgevoerd ontstaat een beeld van de momenten waarop er iets gebeurt voor dat bepaalde beeldscherm. Zet de gebeurtenissen eens uit op een tijds-as, inklusief de POLL's. De tijd tussen een blok heen en een blok terug is de responsetijd. De tijd tussen een blok terug en een blok heen is de invoerrepetitietijd. Als er in die tijd geen of weinig POLL's optreden, is het systeem kennelijk overbelast: in de wait- en thinktime moeten een aantal POLL's vallen.
- Konfronteer de computerleverancier met de gevonden cijfers:
  - het aantal I/O's per seconde,
  - het aantal ENTER's per uur en eventueel
  - de POLL-frekwentie.

Hij behoort te kunnen aangeven of de geleverde konfiguratie deze belasting aankan en zo niet, wat er moet gebeuren om de responsetijden akseptabel te maken.

DEEL 6

voor gebruikers

Kleien of houtsnijden.

Kleien onder agogische leiding
met vluchtige resultaten of
methoden leren toepassen die
houtsnijden voor de eerstko-
mende jaren.

Hoofdstuk 61
Mensen, methoden, middelen

61.1 Interaktieve toepassingen

In dit deel voor de gebruikers houden we ons bezig met de aspekten van interaktieve toepassingen die voor gebruikers van belang zijn. Voorafgaand aan de behandeling van vakmanschap, methoden en middelen moeten we aantal termen bespreken. Gebruikers moeten geen automatiseerder worden, dus gaat het niet om technische termen. Het gaat om een aantal nieuwe termen die in de wereld van de gebruikers hun intrede zullen doen. Bij interaktieve toepassingen gaat het om het bedienen van een computer via beeldschermen. Er kunnen ook andere apparaten worden toegepast, maar het beeldscherm is het meest bekende en meest voorkomende middel.
Onder een beeldscherm verstaan we het geheel van T.V.-scherm en toetsenbord. Het woord toepassing of applikatie slaat op de funktie die wordt uitgevoerd, bijvoorbeeld het intoetsen van orders, het opvragen van produktiegegevens. Bij interaktieve toepassingen vinden interakties plaats tussen mens en computer. Wanneer twee mensen interaktief bezig zijn kan dat betekenen dat ze in gesprek zijn, een dialoog voeren. Bij interaktieve toepassingen gaat het om de wisselwerking tussen mens en computer. Ook in dit geval spreekt men van een dialoog, maar dan tussen mens en computer. De mens leest wat op het scherm staat, denkt na en typt iets in, de computer leest wat is ingetypt, verwerkt dat en zet iets op het

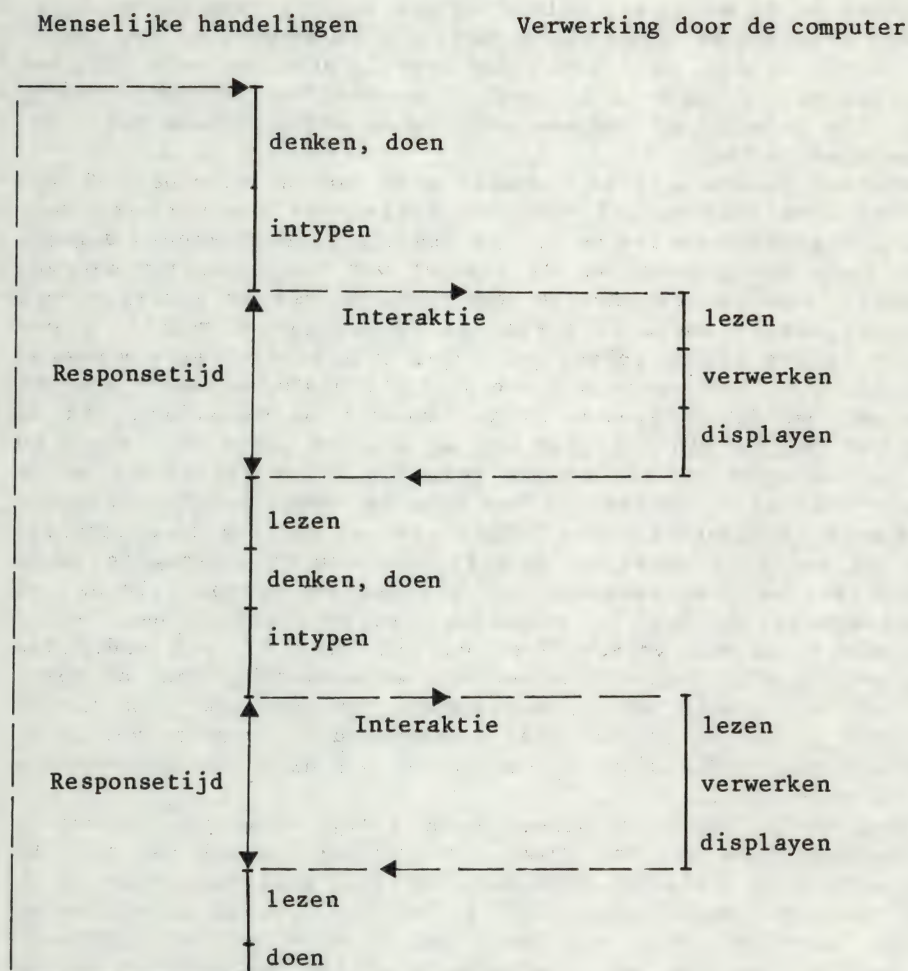


Fig. 61.1 De dialoog tussen mens en computer

scherm, zie Fig. 61.1. Bij interakties gaat het om de wisselwerking tussen de partners. De mens werkt in op de computer en de computer op de mens. Wij zullen in het vervolg onder interaktie alleen verstaan de aktie van de mens die de verwerking van computer start; in Fig. 61.1 komen dus twee interakties voor. Het kan zijn dat om een order in te voeren verscheidene interakties nodig zijn. Als de order is ingevoerd en begint alles opnieuw zoals is aangegeven in Fig. 61.1.

De dialoog tussen mens en computer moet worden ontworpen. De gebruiker moet intypen wat voor hem logisch is, de computer moet geschikt gemaakt worden om dat te begrijpen en te kunnen verwerken. Zoals we spreken van de lay-out van een formulier dat gebruikers invullen, kunnen we ook spreken van de schermlay-out waarmee gewerkt wordt, zie Fig. 61.2. Het aantal regels op een scherm en het aantal tekens per regel zijn beperkt en dus moeten gebruikers en informatie-analist aandacht besteden aan de schermlay-out. De schermlay-out is grotendeels onafhankelijk van de dialoog. Bij de dialoog gaat het om wat een gebruiker intypt en wat de computer op het scherm zet, bij de schermlay-out om de plaats waar het ingetypte en het door de computer getoonde wordt neergezet. Automatiseerders hebben vaak de neiging de schermlay-out het eerst te maken en de gebruiker dan de dialoog te laten raden. Als we even aannemen dat een dialoog bestaat uit een of meer interakties met de computer, kan per dialoog het aantal schermlay-outs nog verschillen. Het invoeren van een order kan bestaan uit drie interakties en een beeldschermlay-out, of zoals het ook wel gezegd wordt, een scherm. Het kan ook zijn dat de computer zoveel informatie moet displayen dat per order verscheidene schermen nodig zijn. Er moeten dan dus drie schermlay-outs worden ontworpen.

Wanneer de dialoog en de schermlay-outs zijn ontworpen, kunnen de automatiseerders aan het werk. Dat doen ze dan ook meestal. De gebruiker moet echter de procedure vertalen naar de nieuwe situatie. Sommige handelingen blijven bestaan, andere vervallen of worden veranderd. Fig. 61.1 geeft ook de werelden van gebruikers en automatiseerders weer. Gebruikers willen vaststellen wat ze moeten intypen en hoe de computer reageert: ze geven in gebruikerstermen aan wat de computer moet doen. De automatiseerders gaan de verwerking door de computer ontwerpen en bouwen. Voor hen is alleen de rechter helft van de dialoog van belang.

Het intypen is voor de gebruiker een gevolg van de vastgestelde procedure en vormt daar een geheel mee. Een procedure waarin gebruik gemaakt wordt van een beeldscherm noemen we een transaktie. Later zullen we zien dat een transaktie wordt vastgelegd op een transaktieschema. Zo'n transaktieschema ziet er in principe uit als Fig. 61.1.

```
Maatschappijnaam x-----------------------------x nummer xx           datum DD-MM-JJ
Programmanaam     x-----------------------------x
Rel.nr.           x-----------------------------x
Periode           x-------------------x
Datum             x-----------------x  x-----x    orderbedrag        zzzzzzzzz9.99
Direkt fakt.  x naleveren  x           valuta                        x  akkoord  x
**************************************************************************************
  Artikelnr.  mag  omschrijving                aantal      prijs  afl.dat.  contr.  akk
1                  x-------------------x
                   x-------------------x
2 **********   x   x-------------------x  ********* zzzz999  JJMMDD  xxxxxx   x
                   x-------------------x  afw. prijs *******  * **.** **.** **.**
3

4

5

Akkoord (90)   volgscherm (96)  opnieuw beginnen (99)  tewijzigen regel   xx
**************************************************************************************
   Verzendadres ** x-----------------------------x            ordernummer   xxxxxx
                   x------------------x
                   x-----------------x                        faktuurnummer xxxxxx
Akkoord (90)  opnieuw beginnen (99)   te wijzigen regel  xx
```

Fig. 61.2 Voorbeeld van een schermlayout.

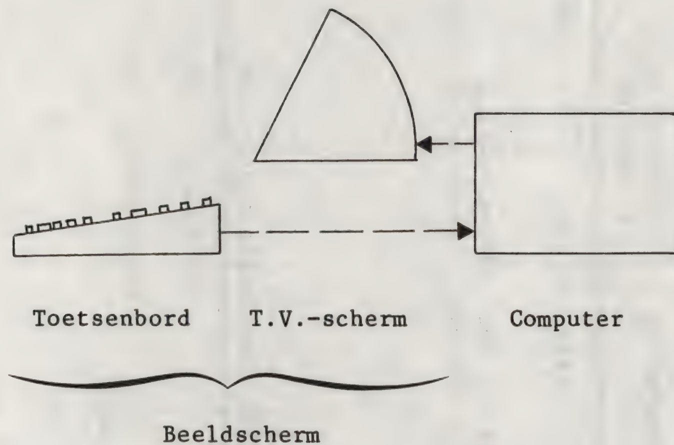


Fig. 61.3 De dialoog met de computer

Tijdens de dialoog moet de gebruiker steeds wachten op de reaktie van de computer. Die wachttijd noemen we de responsetijd. In die tijd is de computer bezig met het verwerken van de ingetypte gegevens. Wachten duurt al gauw lang en dat is erg vervelend: responsetijden zijn erg belangrijk voor de gebruiker. De verwerking door de computer kan zeer ingewikkeld zijn en soms kommuniceert hij nog met een andere computer. Het is moeilijk van te voren te voorspellen hoe lang responsetijden precies worden. Het hele proces van het beheersen van de responsetijden begint echter bij het stellen van korrekte eisen door gebruikers. Bij dialoogsimulatie komen we daar op terug.
Een interaktie ontstaat doordat op een speciale toets wordt gedrukt. Wat er op die toets staat, kan per soort beeldscherm verschillen. De meest voorkomende aanduidingen zijn: F1, ... F10, ENTER, TRANSMIT, SEND, END OF BLOCK, CR of een omhaalpijl. We zullen nu alle besproken begrippen nog even onder elkaar zetten.

- Beeldscherm: T.V.-scherm en toetsenbord
- Schermlay-out of scherm: de indeling van het T.V.-scherm
- Dialoog: kommunikatie tussen mens en computer. De mens typt in via het toetsenbord en leest op het scherm, de computer leest van het toetsenbord en zet informatie op het scherm
- Interaktie: In het algemeen de wisselwerking tussen mens en

computer. Hier het starten van de verwerking door de computer
- Responsetijd: de tijd die nodig is voor lezen, verwerken en displayen door de computer
- Transaktie: de procedure aan het beeldscherm
- Toepassingen of applikaties: computerfunkties zoals het verwerken van orders, het registreren van bezoekers, het besturen van een produktieproces.

61.2 Vakmanschap

Wanneer het gaat om het vakmanschap van de gebruiker, zullen we eerst onderscheid moeten maken tussen de verschillende gebruikers. Een loketbediende kent de procedure aan het loket, hij vult formulieren in, hij verricht boekingen, en legt de transaktie vast. Hij hoeft niets te weten van de verwerking van de door hem vastgelegde gegevens. De afdelingschef moet weten hoe de dag is verlopen en heeft bepaalde totaalcijfers nodig per loket.
Nog een niveau hoger worden de afdelingscijfers, samen met andere gegevens, verwerkt tot gegevens die van belang zijn op dat niveau. In de geautomatiseerde situatie kunnen al deze gebruikers gebruikmaken van een beeldscherm. Allen hebben dan te maken met een procedure om met het beeldscherm te werken. Meestal is er echter een groot verschil in inzicht in wat er met de gegevens gebeurt. Die situatie verschilt dus nauwelijks van de handmatige situatie. De loketbediende moet zorgvuldig zijn staten bijhouden, maar de verwerking van die staten is vaak een heel ander terrein. Hij heeft de formulieren dan ook niet ontworpen. Hoewel hij niet veel invloed heeft op wat er moet worden vastgelegd, heeft hij meestal wel een duidelijke mening over hoe het vastleggen het handigst kan gebeuren. De ontwerpers van de formulieren zitten nu eenmaal niet achter zijn loket. En ook al zouden ze daar ooit gezeten hebben, dan is het nog verstandig om bij het ontwerpen van formulieren de mensen in te schakelen die ermee moeten werken. In de geautomatiseerde situatie blijft het geschetste beeld hetzelfde. We spreken van eindgebruikers en van gebruikers. Gebruikers in een bedrijf zijn eigenlijk alle nietautomatiseerders. Eindgebruikers bevinden zich op uitvoerend niveau, aan het front, de grenslijn van het bedrijf. Zij aksepteren orders, telefoneren met klanten, maken inkooporders enzovoort. Voor het ene beroep is MAVO een goede opleiding, voor het andere is een universitair niveau noodzakelijk. In het ene bedrijf is het uitvoerend niveau heel sterk gescheiden van de verwerking, in het andere bedrijf wordt alles door een persoon gedaan. Aangezien, zoals gezegd, iedereen te maken krijgt met een procedure aan het beeldscherm en we in deze hoofdstukken praten over het ontwerpen van die procedure, maken we geen verschil tussen soorten gebruikers. Gaat het

over de verwerking door de computer dan maken we wel onderscheid. Dan worden met eindgebruikers de mensen bedoeld die zich niet verdiepen in de verwerking van de door hen ingevoerde gegevens. Zij zorgen voor het korrekt invoeren van de gegevens en de afhandeling van de brondokumenten. Voor alle toekomstige beeldschermgebruikers moeten dus een of meer procedures worden ontworpen. Die procedure moet aansluiten op het overige, handmatige werk, maar moet ook voorzien in de behoeften van de gebruiker. In een computersysteem ligt een schat aan gegevens opgeslagen. De gebruikers kunnen zelf het beste aangeven wat ze willen: nieuwe mogelijkheden in het werk van de gebruikers moeten door de gebruikers bedacht worden. Dat het meestal niet gebeurt komt doordat gebruikers in eerste instantie geen idee hebben van wat een computer eigenlijk is en als de automatisering is ingevoerd blijkt het bijna onmogelijk nog andere mogelijkheden toe te voegen. Niet dat het technisch niet zou kunnen, maar het duurt lang, kost veel of krijgt een lage prioriteit. De methoden die in de volgende paragraaf worden behandeld maken het nu juist mogelijk om met een beeldscherm te werken voordat het computersysteem is ingevoerd. Daardoor neemt de kans toe dat de gebruiker in de ontwerpfase met kreatieve mogelijkheden komt, die dan nog eenvoudig in het geheel zijn in te passen. Het gaat erom het vakmanschap van de gebruiker te koppelen aan dat van de automatiseerder. De gebruiker komt met een idee, de automatiseerder toont mogelijke oplossingen, laat de gebruiker ermee werken en zo komen ze samen tot een afweging van de voor- tegen de nadelen. Dat werkt alleen maar als de automatiseerder beschikt over de middelen die een snelle realisering van een idee mogelijk maken. De gebruiker blijft dus vakman op zijn gebied en hij hoeft geen automatiseerder te worden. Hij moet straks wel een beeldscherm bedienen. Hij mag dus best het een en ander afweten van beeldschermen en de bediening. In veel bedrijven gebeurt dat nadat de beeldschermen zijn geplaatst. Als we het hebben over gebruikers als mede-ontwerpers dan moet de kennis omtrent de bediening aanwezig zijn tijdens de ontwerpfase van het systeem. Dat betekent dat de gebruiker een algemene kennis moet hebben van het gebruik van beeldschermen en van de kommunikatie met de automatiseerders. Dat is dus iets anders dan leren hoe een computer werkt of hoe je een BASIC-programma schrijft. Het gaat om een opleiding, als (15), waarin gebruikers leren hoe ze met een beeldscherm moeten omgaan, hoe ze een bijdrage kunnen leveren in het ontwerpproces en hoe het hele automatiseringsgebeuren is gefaseerd, zodat ze de vinger aan de pols kunnen houden.
De gebruiker hoeft geen automatiseerder te worden, de automatiseerder geen gebruiker. Dat is geen probleem, wanneer beide groepen goed zijn voorbereid op een kommunikatie die methodisch is vastgelegd en tot kontroleerbare resultaten leidt. Het is te sim-

pel om vast te stellen dat automatisering banen kost. Dat is waar, wanneer we niet verder komen dan het automatiseren van de bestaande situatie. Automatisering kan ook betekenen dat het bedrijf effektiever funktioneert met hetzelfde aantal mensen. Die mogelijkheden kunnen alleen grondig worden onderzocht, wanneer het vakmanschap van de gebruiker gekoppeld wordt aan het vakmanschap van de automatiseerder. De gesprekspartner voor de gebruiker is de informatie-analist. In het deel voor de informatie-analist is aangegeven, waarin zijn vakmanschap moet bestaan, wat het ontwerpen van de procedure aan het beeldscherm betreft.

61.3 Methoden en middelen

Een methode is een uitgekristalliseerde, gestandaardiseerde en goed gedokumenteerde manier van werken, waarbij eventueel gebruik gemaakt wordt van middelen of gereedschappen, tools. Bij methoden voor het realiseren van de inbreng van gebruikers in het ontwerpproces gaat het om methoden die door automatiseerders en gebruikers zijn geselekteerd vanwege de resultaten die ze opleveren. Het interviewen van gebruikers is, in deze zin, geen methode. Iedereen doet dat op zijn eigen manier: noch voor de diepgang, noch voor de manier waarop de resultaten voor gebruiker en automatiseerder worden vastgelegd bestaan standaards. Hetzelfde geldt voor het inschakelen van gebruikers bij het ontwerpproces. Wanneer met de gebruiker een map met tientallen schermlay-outs is doorgenomen, is, volgens sommigen, de gebruiker ingeschakeld tijdens het ontwerp. Een werkwijze die is uitgekristalliseerd, is al vele malen toegepast en bijgeschaafd door de praktijk. Alle valkuilen zijn bekend en erin vallen wordt voorkomen.
De manier van werken wordt gestandaardiseerd omdat

- Gebruikers en automatiseerders de methode hebben getoetst op z'n bruikbaarheid,
- de resultaten kontroleerbaar zijn,
- niet iedereen de vrijheid heeft het wiel opnieuw uit te vinden, daarbij de kans lopend dat het helemaal geen wiel wordt.

De aanpak moet goed gedokumenteerd zijn zodat de te bereiken resultaten precies vastgesteld kunnen worden en nieuwe medewerkers snel ingewerkt kunnen worden in de gekozen werkwijze.
Bij middelen kan men denken aan in te vullen formulieren maar ook aan computers. In zijn algemeenheid gaat het om gereedschappen die ontstaan zijn tijdens de ontwikkeling van de methoden. Als een werkwijze wordt gestandaardiseerd, dus vele malen op dezelfde manier wordt uitgevoerd, is het zinvol te zoeken naar middelen die de toepassing van de methode versnellen of vergemakkelijken. Kortom, wanneer gekozen is voor methoden op basis van hun resultaten voor de gebruikers, dan biedt de konsekwente uitvoering er-

van een stuk zekerheid voor die gebruikers. Daarom is het beter dat gebruikers de tijd nemen zich te verdiepen in dit soort methoden, dan zich bezig te laten houden met allerlei vage bewustwordingsprocessen, het aanleren van kommunikatieve sociovaardigheden, waarvan een maand later niemand het effekt kan aanwijzen en het voeren van diskussies over sociale aspekten van de automatisering, die nooit konkreet worden.
We behandelen twee methoden: dialoogsimulatie en Transaktie analyse.
Dialoogsimulatie is een methode die het mogelijk maakt, zelfs voordat de computer is aangeschaft, met het beeldscherm te werken op iedere soort werkplek. Transaktie analyse maakt het mogelijk de gebruikerssituatie om te rekenen naar cijfers over bijvoorbeeld aantallen beeldschermen en uren per beeldscherm. Sommige automatiseerders hebben een hekel aan methoden: ze voelen het als een harnas dat het werken bemoeilijkt. Ze willen de vrijheid om op hun eigen manier te werken. Het is duidelijk dat die eigen manier van werken het einde betekent van de zekerheid van de gebruikers ten aanzien van de inspanning die zij moeten leveren en de te verwachten dokumenten, resultaten en konklusies.
Er bestaat een 20 tot 30 jaar ervaring in automatiseren. De laatste 10 jaar hebben zich twee belangrijke veranderingen voorgedaan: computers worden uitgerust met beeldschermen en veel kleine bedrijven gaan over tot automatiseren. Men heeft getracht de aanpak van de automatisering vast te leggen in systeemontwikkelingsmethoden: het hele gebeuren vanaf de eerste globale plannen tot en met de invoering, zijn in kaart gebracht al dan niet kompleet met methoden en middelen. Voordat beeldschermen werden gebruikt, braakten computers enorme hoeveelheden papier uit: talloze lijsten, overzichten, tabellen enzovoort. Op een of andere wijze was de gebruiker natuurlijk betrokken bij de gegevens die op papier verschenen en bij de lay-out van de lijsten en overzichten. Zo gauw de gebruiker had aangegeven wat uit de computer moest komen en hoe dat op papier moest verschijnen was voor de automatiseerder de kommunikatie met de gebruiker afgesloten: een simpele, eenduidige kommunikatie.
In alle systeemontwikkelingsmethoden kwamen dan ook steevast de twee aspekten voor: het interviewen van gebruikers en het ontwerpen van in- en uitvoer. Tijdens de interviews vertelt de gebruiker hoe hij nu werkt en hoe zijn dokumenten eruit zien. Afhankelijk van de funktie van het computersysteem, was het eenvoudig vast te stellen welk deel van de gegevens waar de gebruiker mee werkte, in de computer moesten worden ingevoerd, door ze op ponskaarten vast te leggen. De gebruikers die met de resultaten van de verwerking door de computer moesten omgaan, mochten zeggen hoe ze die resultaten op papier wilden zien. Daarmee was de kom-

munikatie tussen gebruikers en automatiseerders afgesloten. Voor al het andere zorgden de automatiseerder en na een jaar of langer begonnen de pakken papier binnen te stromen. De gebruikers zochten hun gegevens op, bestudeerden resultaten, trokken hun konklusies of legden de stapels papier direkt bij het ronde archief.
Voor veel automatiseerders heeft de komst van het beeldscherm niet veel veranderd aan hun manier van werken. De gegevens worden nu via het toetsenbord van een beeldscherm ingevoerd, in plaats van het invoeren met ponskaarten, de resultaten worden niet meer op papier afgedrukt, maar verschijnen op het scherm. Op de meeste systeemontwikkelingsmethoden heeft het beeldscherm dan ook weinig invloed gehad. Er wordt nog steeds geinterviewd en naast de papierlay-out moet nu ook de schermlay-out met de gebruikers worden doorgenomen.
Ook al geeft iedere automatiseerder toe dat er toch wel een verschil bestaat tussen het bladeren in een pakpapier en het interaktief werken met een computer via een beeldscherm, er zijn weinig methoden ontwikkeld die de gebruiker tijdens de ontwerpfase laten ervaren wat het werken met een beeldscherm nu precies inhoudt. Dat is heel wat anders dan ze een schermlay-out laten aksepteren. Het is gemakkelijk een gebruiker akkoord te laten gaan met een schermlay-out, als je niet vertelt dat hij soms twintig van deze schermlay-outs door moet bladeren voor hij de juiste informatie heeft en dat er tussen iedere twee schermen een wachttijd zit van 5 seconden. Het zou niet de eerste keer zijn dat gebruikers de schermlay-out aksepteren, maar de hele procedure achteraf onakseptabel vinden. Maar dan is het hele systeem ontwikkeld en klaar, dan is er geinvesteerd, dan kosten wijzigingen goud en het niet gebruiken van bepaalde toepassingen is een vorm van kapitaalvernietiging.
Het probleem van de automatiseerders is, dat ze de gebruikers pas kunnen laten zien hoe een beeldscherm funktioneert in hun werksituatie als de computer is aangeschaft en het systeem is ontworpen en gebouwd. Pas dan kan de gebruiker het systeem beoordelen en voor een kreatieve gebruiker is het dan meteen ook te laat voor suggesties.
In de volgende hoofdstukken bespreken we de methoden en de middelen om
- de gebruikers in te schakelen bij het ontwerpproces,
- de gebruiker tijdens het ontwerpproces te laten ervaren hoe de uiteindelijke geautomatiseerde situatie op zijn werkplek er uitziet,
- de gebruiker tijdens het ontwerpproces te konfronteren met cijfers over beeldschermen per afdeling, de bezetting van die beeldschemren, het aantal beeldschermuren per dag, kortom, de sociale aspekten in cijfers.

61.4 Projektaanpak

Met het woord projektaanpak of systeemontwikkelingsmethode wordt bedoeld de manier van werken van de automatiseerders. Zij pakken op een bepaalde manier de automatisering aan. Soms is de manier van werken goed vastgelegd in de handboeken van de automatiseerders, soms ligt er niets vast. In alle gevallen wordt er natuurlijk op een bepaalde manier gewerkt. Een computer gaat niet uit zichzelf een administratieve funktie uitvoeren. Iedereen die wel eens iets bedacht heeft, kan vaststellen dat er in het denkproces altijd dezelfde fasen voorkomen:
- het vaststellen van het probleem of de behoefte,
- het bedenken van een aantal oplossingen,
- het uitwerken van de beste oplossing,
- de realisering, de bouw,
- het testen en beoordelen van het produkt,
- het gebruik en de evaluatie.

Bij de invoering van computersystemen gaat het in principe net zo. Iedere fase kan echter zo ingewikkeld zijn dat er een groep specialisten voor nodig is om hem uit te voeren. We zullen nu de bovengenoemde fasen weergeven in de termen van de automatisering. Het geheel van de fasen heet de projektaanpak, de fasen zijn:
- de analyse van de huidige situatie door informatie-analisten
- het maken van het logisch ontwerp door informatie-analisten
- het bouwen van het systeem door programmeurs
- het uitvoeren van systeemtests door programmeurs en systeemontwerpers
- de akseptatietest en de invoering door informatie-analisten en systeembeheerders

Er zou over het aantal fasen nog heel wat te zeggen zijn, maar in het kader van het onderwerp, zijn dit de belangrijkste fasen.
Het is duidelijk dat een projektaanpak bedoeld is voor automatiseerders. Natuurlijk komen er wel gebruikers in voor: die mogen tijdens interviews vertellen hoe ze nu werken en vervolgens wachten tot de beeldschermen worden binnengereden. Soms worden ze tijdens het logisch ontwerp nog "ingeschakeld". Ze mogen dan schermlayouts bekijken en kommentaar geven. Soms blijkt dan dat de automatiseerders moeilijk af te brengen zijn van hun mening over wat er wel en niet op een scherm moet staan. Dat kan komen doordat ze al met de bouw zijn begonnen: wijzigingen tijdens de bouw zijn erg vervelend.
Wanneer de beeldschermen zijn binnengereden volgt de akseptatietest, die in de praktijk slechts de mogelijkheid biedt ja te zeggen tegen wat er ontwikkeld is. De gebruikers moeten nu nog even leren werken met het systeem, dat wil zeggen, zich leren aanpassen aan het systeem. Daarmee is het projekt gerealiseerd: de ge-

bruiker weet hoe hij de eerstvolgende jaren moet werken en de automatiseerders gaan vol enthousiasme beginnen aan het volgende projekt, als dat niet al lang gestart is. De gebruiker wordt dus af en toe even gebruikt in de projektaanpak van de automatiseerders. Waar behoefte aan is, is een projektaanpak voor gebruikers. Gebruikers moeten precies weten welke fasen er doorlopen moeten worden, wat er van hen verwacht wordt, welke tijd ze daarvoor moeten reserveren, welke dokumenten de automatiseerders moeten opleveren, enzovoort.
De projektaanpak voor gebruikers moet synchroon lopen met de projektaanpak van de automatiseerders. Ten opzichte van de reeds genoemde fasen zullen we aan het begin nog een fase toevoegen: het vooronderzoek. Meestal gaat het hier om de grote lijnen van het projekt. Er wordt globaal gekeken naar bestaande problemen en knelpunten, mogelijke oplossingen en kosten. Vaak is dit voor de gebruikers de vervelendste fase. Er wordt gesproken en vergaderd door het management en staffunktionarissen over automatiseringen en niemand weet iets zeker. Tijdens die fase is het inderdaad zo dat niemand iets weet. Er worden nog geen beslissingen genomen, er wordt overwogen. Voor de gebruikers die niet bij dit afwegen zijn betrokken, hoeft er ook geen enkele angst te bestaan als ze goed weten wat er na dat vooronderzoek nog allemaal moet gebeuren en welke inbreng ze daarbij hebben volgens de overeengekomen methoden. Dat werkt alleen als ook de gebruikers zich strikt houden aan hun projektaanpak en de relatie met de projektaanpak van de automatiseerders goed bewaken. Voor we de fasen van de projektaanpak van de gebruikers bespreken merken we nog het volgende op.
- Soms zullen automatiseerders zeggen dat de beschreven fase-indeling niet meer gehanteerd wordt vanwege de toepassing van moderne methoden als prototyping en vierde-generatietalen. Hoe een systeem ook ontwikkeld wordt, de genoemde fasen zijn altijd terug te vinden en de methoden dialoogsimulatie en Transaktie analyse kunnen altijd worden uitgevoerd.
- De projektaanpak voor automatiseerders is niet in alle bedrijven dezelfde. De grote lijnen en de indeling in fasen komen altijd voor, maar de werkwijze per fase of de manier waarop gegevens worden vastgelegd kan per bedrijf verschillen.
- Niet iedere projektaanpak geeft precies aan volgens welke methode er gewerkt moet worden en welke gereedschappen daarbij gebruikt moeten worden. Soms wordt alleen aangegeven wat er moet gebeuren, maar men kan zelf bepalen hoe men het doet.
- Soms wordt gekozen voor een systeemontwikelingsmethode. Dan gaat het om een konsistent geheel van methoden en middelen, dat een aantal van de genoemde fasen dekt, maar bijna nooit alle. In het kader van het onderwerp voert het te ver om alle aspekten, voor alle fasen van de projektaanpak van de gebruikers te behan-

delen. We leggen het aksent op fasen, methoden en middelen die te maken hebben met de inbreng van gebruikers bij het ontwerp van interaktieve toepassingen. Het gaat daarbij zowel om kwalitatieve inbreng, wat moet er gebeuren, als kwantitatieve inbreng, hoe vaak moet het gebeuren. In deze zin zullen we de verschillende fasen behandelen. Fig. 61.4 brengt het geheel in beeld.

- Het vooronderzoek.

In deze fase gaat het om de grote lijnen. Er is nog niet bekend wat er zou kunnen worden geautomatiseerd, laat staan of dat met beeldschermen moet of op een andere manier. Het management geeft alleen de grote lijnen aan en daarom zullen eindgebruikers niet betrokken zijn bij dit onderzoek. Als in een P4/P5/P6-omgeving (zie flap), de aanschaf van de computer wordt uitgesteld tot na het logisch ontwerp, hoeft ook niemand belang te stellen in de overwegingen van het taktisch management. Dat werkt alleen als door het taktisch gebruikersmanagement gekozen is voor methoden die de inbreng van de gebruikers garanderen. Die methoden zullen we aangeven bij de fase logisch ontwerp. Daar zal blijken dat gebruikers, geheel onafhankelijk van de aanwezigheid van een computer, al kunnen werken met beeldschermen en dus de uiteindelijke procedure kunnen ontwerpen. Dat kan in principe ook al tijdens het vooronderzoek, maar het is niet verstandig het vooronderzoek te bemoeilijken met de details per werkplek. Toch kan het zijn dat een direktie tijdens het vooronderzoek al een nauwkeurige analyse wil hebben van een bepaald aspekt van de automatisering. Dat zou het netwerk kunnen zijn, dat nodig is om bijvoorbeeld een groot aantal beeldschermen op allerlei vestigingen te verbinden met de computer. Als de kosten van zo'n netwerk maatgevend zijn voor het doorgaan van de automatisering, zou men kunnen besluiten om tijdens het vooronderzoek dat aspekt van de automatisering alvast nauwkeurig te onderzoeken. Met de methoden die we zullen beschrijven is dat mogelijk. In zo'n geval wordt het interaktieve gebeuren per werkplek in kaart gebracht. Gebruikers en informatie-analisten ontwerpen transakties en laten collega-gebruikers ermee werken om ze te beoordelen. Hoe "live" de automatisering dan ook wordt, het blijft plaatsvinden in de fase Vooronderzoek en aan het eind daarvan kan best besloten worden af te zien van de hele automatisering.

Afgezien van de uitzonderingen is het vooronderzoek niet het moment om procedures aan het beeldscherm te ontwerpen. Informatie-analisten zullen wel kontakt opnemen met gebruikers om bijvoorbeeld vast te stellen welke dokumenten per werkplek worden behandeld, welke gegevens erop zijn vermeld, waar ze vandaan komen en waar ze naar toe gaan. Zeker als de informatie-analisten ingehuurde krachten zijn, hebben ze deze vorm van kommunikatie nodig om een inzicht te krijgen in de bedrijfsprocessen. Als methoden

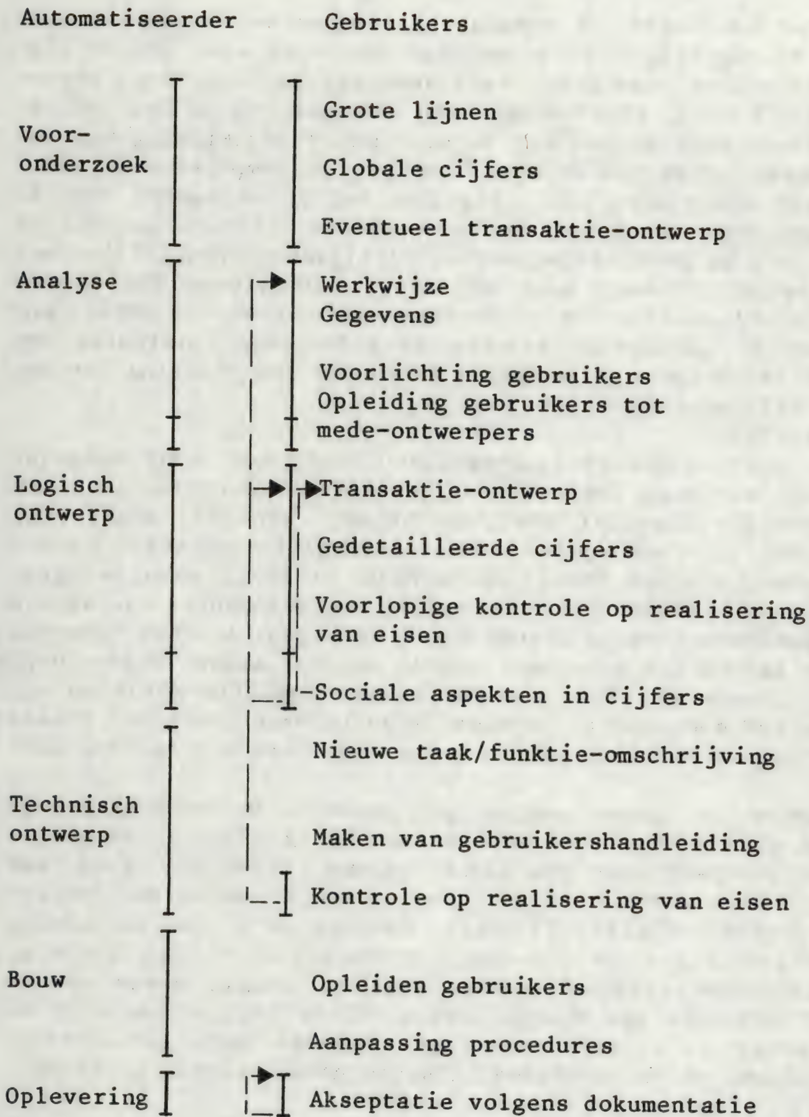


Fig. 61.4 Projektaanpak en gebruikers.

wordt in het vooronderzoek meestal het houden van interviews genoemd. In M1-omgevingen is er meestal ook niet meer van te zeggen. Ook de manier waarop de resultaten van de interviews worden vastgelegd, is vrij. In M3-omgevingen gaat men ook in het vooronderzoek steeds meer methodisch te werk met gebruikmaking van geautomatiseerde hulpmiddelen voor kontrole op kompleetheid, vaststelling van knelpunten (20). Afgezien van de voordelen voor de automatiseerders, heeft een methode op dit terrein duidelijke voordelen voor de gebruikers. Geen vrijblijvende, nauwelijks voor te bereiden interviews, maar standaard formulieren die moeten worden ingevuld samen met de informatie-analist en een snelle indikatie van de konklusies waartoe de verstrekte informatie zou kunnen leiden. Dergelijke methoden maken ook een planning van de benodigde tijd mogelijk.

- De analysefase.

In sommige systeemontwikkelingsmethoden maakt het analyseren van de bestaande situatie deel uit van het logisch ontwerp. In feite is dat voor de gebruiker niet van belang, want hij merkt heel duidelijk dat op een gegeven moment informatie-analisten de bestaande procedures en bedrijfsprocessen in kaart gaan brengen. Meestal kunnen gebruikers niet overzien hoe hun manier van werken er in de geautomatiseerde situatie uit zal zien. Als het invoeren van orders zal worden geautomatiseerd, zal het aantal orders hopelijk niet veranderen. Maar als het straks mogelijk wordt op elk moment via een beeldscherm informatie op te vragen over de status van een order is moeilijk aan te geven, hoe vaak dat per dag zal gebeuren.

Het is daarom van belang dat de gebruiker en de informatie-analist samen goed vaststellen waar ze mee bezig zijn. In de analyse-fase is de vraag naar het aantal orders legaal, de vraag naar het aantal keren per dag dat een nieuwe, nog te ontwerpen toepassing zal worden gebruikt, illegaal. Pas als de nieuwe toepassing samen met gebruikers is ontworpen tijdens het logisch ontwerp, moet de gebruiker uitspraken doen over het aantal keren dat de toepassing gebruikt zou kunnen worden. In de analysefase moet de gebruikersorganisatie over mensen met voldoende materiedeskundigheid beschikken om de bestaande situatie nauwkeurig in kaart te brengen.

- Het logisch ontwerp.

Dit is voor de gebruikers de belangrijkste fase. Aan het eind van het vooronderzoek is in principe de beslissing genomen door te gaan. Tot nu toe zijn gebruikers hoogstens betrokken geweest bij het in grote lijnen beschrijven van het bedrijfsgebeuren. Bij het logisch ontwerp gaat het om de funktionele specifikatie van wat de computer moet doen. Men spreekt van het logisch of funktioneel ontwerp, omdat het nog niet gericht is op een bepaalde computer.

Het gaat om principe-ontwerpen. Een procedure aan het beeldscherm kan ontworpen worden omdat alle beeldschermen in principe hetzelfde kunnen. Kleine afwijkingen zijn pas bespreekbaar in het technisch ontwerp. In P3-omgevingen is de situatie wat dat betreft ideaal, omdat dan precies bekend is welke beeldschermen en welke computer gebruikt zal worden.
Eerst worden alle gebruikers voorgelicht over de methoden die zullen worden toegepast. Informatie-analisten, die zijn opgeleid via kursussen als (16), kunnen een presentatie houden over methoden die de gebruikers betreffen.
Vervolgens moeten gebruikers worden opgeleid om op te kunnen treden als mede-ontwerper, te begrijpen wat er in de verschillende fasen moet gebeuren en de benodigde tijd te kunnen schatten. In kursussen als (15) wordt daaraan inhoud gegeven.
Tijdens het logisch ontwerp wordt onder andere de situatie per werkplek in kaart gebracht. Welke aktiviteiten en procedures worden daar uitgevoerd en hoe zouden die geautomatiseerd kunnen worden? Nu zal er ook gevraagd worden naar kwantiteiten: hoeveel orders per dag, hoeveel orderregels per order, welke pieksituaties bestaan er, enzovoort. Het is duidelijk dat het invoeren van een order tijd kost. Het aantal orders dat per dag moet worden verwerkt, bepaalt het aantal benodigde beeldschermen voor order entry. Het totaal van de beeldschermen van alle afdelingen bepaalt de grootte van het computersysteem en de kosten van het geheel. Een van de methoden die behandeld wordt maakt het mogelijk om tijdens het logisch ontwerp bijvoorbeeld te begroten hoeveel beeldschermen er nodig zijn. Dit kan door de automatiseerders vertaald worden naar gevolgen voor het computersysteem, maar door de gebruikers naar het benodigd aantal mensen om de beeldschermen te bedienen. Vandaar dat cijfers belangrijk zijn.
Tenslotte vindt het belangrijkste plaats en dat is voor de interaktieve toepassingen het transaktie-ontwerp. Een transaktie is een procedure aan een beeldscherm. Het ontwerp daarvan bepaalt het gezicht van de automatisering. Tijdens dat ontwerp worden de eisen gesteld. Het belangrijkste aspekt van de methoden die tijdens het logisch ontwerp kunnen worden toegepast, is dat voor de gebruikers de uiteindelijke, nog te bouwen situatie opgezet en door hen beoordeeld kan worden. De sociale aspekten van de automatisering worden nu per werkplek konkreet en kunnen in cijfers worden uitgedrukt. De pijlen in Fig. 61.4 geven aan dat er nog een weg terug mogelijk is voor de gemaakte ontwerpen. Wanneer gebruikers namelijk niet gelukkig zijn met een ontwerp, moeten ze om alternatieven vragen of eisen dat de bestaande situatie nauwkeuriger wordt onderzocht.
Het logisch ontwerp is niet afhankelijk van de computer. Dat betekent dat de realisering van bijvoorbeeld responsetijdeisen

slechts in principe kan worden aangegeven. Bij grote afwijkingen moet tijdens het logisch ontwerp reeds naar andere oplossingen worden gezocht. Van andere afwijkingen zullen de auotmatiseerders moeten aangeven wat de kansen zijn dat tijdens het technisch ontwerp de zaak er goed uit komt te zien. Tijdens het logisch ontwerp zijn de te verwachten technische resultaten bekend. Het einde van het logisch ontwerp betekent tevens het einde van de gebruikersinbreng. Niets is zo slecht voor de voortgang van een projekt en de motivatie van de automatiseerders als gebruikers die blijven komen met nieuwe suggesties en plannen, soms nog tijdens de bouw!

- Het technische ontwerp.

Tijdens deze fase werken de automatiseerders aan de vertaling van het logisch ontwerp naar het technisch ontwerp. In principe dus iets waar de gebruiker niets van hoeft te merken. Toch is er een punt van wezenlijk belang en dat is het einde van het technisch ontwerp. Dat is het laatste moment waarop nog kan worden ingegrepen. Een laatste kontrole op de realisering van de gestelde eisen is hier noodzakelijk. Wie dit punt ongebruikt voorbij laat gaan verliest alle recht van spreken. Hier mogen automatiseerders hun vakmanschap bewijzen door te garanderen dat er aan de eisen van de gebruikers wordt voldaan. Mocht dat om allerlei redenen niet kunnen dan biedt het technisch ontwerp nog altijd de mogelijkheid om terug te keren. Er bestaan een paar mogelijkheden:

- de gebruiker annuleert bepaalde transakties,
- de gebruiker aksepteert en stelt de eisen bij,
- er wordt een alternatief ontworpen,
- een deel van de analyse van de handmatige situatie wordt opnieuw uitgevoerd inclusief het daarbij behorende logisch en technisch ontwerp.

Aanpassingen tengevolge van het technisch ontwerp worden verwerkt in de gebruikersdokumentatie en daarmee kan, terwijl de automatiseerders verder ontwerpen of bouwen, gestart worden met het opzetten van de gebruikershandleiding en het maken van de nieuwe taak/funktie-omschrijvingen. Aanpassingen in de organisatie kunnen worden voorbereid. Naarmate de overdracht nadert kunnen gebruikers al worden opgeleid in het bedienen van de beeldschermen. Dat versnelt de overdrachtstest aanmerkelijk.

- De akseptatietest.

Voorzover het woord akseptatie slaat op het aksepteren van iets nieuws, iets onbekends, is het nu overbodig geworden: de akseptatie heeft plaatsgevonden tijdens het logisch ontwerp. Nu wordt het bestelde systeem opgeleverd en overgedragen. Gebruikers kontroleren aan de hand van hun gegevens of het systeem werkt zoals ze het besteld hebben. Wat het voor hen en hun werk betekent weten ze al lang.

- De invoering en evaluatie.
Doordat de gebruikers tijdens het logisch ontwerp al gewerkt hebben met de beeldschermen is iedereen ook goed duidelijk wat de nieuwe manier van werken gaat worden. Dat betekent dat na het logisch ontwerp de invoering in de organisatie al voorbereid kan worden. De evaluatie van werkende systemen is een kontinue proces. Op gezette tijden moet een hoeveelheid geaksepteerde wijzigingsvoorstellen worden vertaald naar een nieuwe release van de software. Ook alle suggesties van gebruikers die na het logisch ontwerp nog boven water zijn gekomen behoren te worden meegenomen als wijzigingsvoorstellen. Voor de gebruikers is de tijd voor het doen van voorstellen voorbij, als het logisch ontwerp voorbij is. Hoe goed die voorstellen ook zijn, in principe worden ze pas verwerkt in het tweede release. Als de automatiseerders bereid zijn ze alsnog mee te nemen, is dat meegenomen, maar gebruikers behoren dat niet meer te eisen, tenzij ze een uitloop van het projekt en hogere kosten aksepteren evenals de kans op demotivatie bij de automatiseerders. Het is nu eenmaal niet leuk binnen een paar maanden 5 keer opnieuw te moeten beginnen met een technisch ontwerp!
Methoden moeten passen binnen een projektaanpak en aansluiten bij aangrenzende methoden. In de delen voor automatiseerders is aangegeven dat de te behandelen methoden die inbreng van de gebruikers mogelijk maken, zijn in te passen in elke projektaanpak in elk bedrijf.

61.5 Kommunikatie met automatiseerders

In deze paragraaf gaat het niet over kommunikatieve vaardigheden in het algemeen maar om de kommunikatie tussen personen in het kader van methoden die we behandelen. Het gaat om:
- wie kommuniceert met wie?
- waarover kommuniceren zij?
- wanneer kommuniceren zij?

In het algemeen is binnen de automatiseringsafdeling de informatie-analist degene die kommuniceert met gebruikers tijdens het ontwerp van computersystemen. Vaak is het een automatiseringsdeskundige met een hoeveelheid materiekennis, soms is het iemand uit de gebruikersorganisatie met automatiseringskennis. Ingehuurde krachten behoren meestal tot de eerste groep. In het deel voor de informatie-analisten is beschreven wat tot het vakmanschap van een informatie-analist behoort, wil hij goed voorbereid zijn op de kommunikatie met de gebruiker.
Wie de gebruikers zijn die betrokken worden bij het ontwerpproces is vaak moeilijker aan te geven. Bij honderd gebruikers wordt het een probleem ze allen voor 100% te betrekken bij het ontwerppro-

ces. Bij de behandeling van de methoden zal duidelijk worden dat het ontwerpproces wat de gebruikers betreft, in twee fasen is te verdelen. In de eerste fase ontwerpt een groepje van enkele gebruikers samen met de informatie-analist de transakties. Die worden vervolgens aan grotere groepen gebruikers gedemonstreerd. Meestal wordt heel snel duidelijk of men met het ontwerp op de goede weg is of niet. Ook de mensen met goede suggesties krijgen nu de gelegenheid een bijdrage te leveren. Mocht er erg veel kommentaar komen dan betekent dat dat de keuze van de groep gebruikers die bij het ontwerp was betrokken, niet optimaal is geweest, maar het ontwerp kan snel worden aangepast en weer worden gedemonstreerd. In de praktijk blijkt dat het kommentaar meestal over details gaat. In ieder geval moeten de verschillende groepen gebruikers in overleg met hun management vaststellen wie er bij de twee fasen worden betrokken. Degenen die betrokken worden bij de ontwerpfase moeten zeker voor de kommunikatie met de informatie-analist worden opgeleid (15). Dan weten ze van te voren wat het betekent om medeverantwoordelijk te zijn voor het ontwerp. Ze weten dan welke methoden gebruikt worden, welke dokumenten gemaakt moeten worden en ze hebben een idee van de te reserveren tijd.

Een belangrijk aspekt van de methoden die we behandelen is dat gebruikers tijdens het logisch ontwerp aan den lijve ondervinden wat het betekent met een beeldscherm te werken en dat er cijfers komen over de toekomstige situatie. Daarbij zal duidelijk worden dat het aantal beeldschermen op een afdeling kan worden berekend door informatie-analisten, maar dat die berekening uiteindelijk is gebaseerd op cijfers van de gebruikers. In de praktijk zijn cijfers niet altijd eenvoudig te geven. De gebruiker mag eisen van de automatiseerders dat ze voor verschillende schattingen hun berekeningen uitvoeren. Op die manier krijgt de gebruiker gevoel voor de konsekwenties van onnauwkeurigheden in zijn kwantitatieve gegevens. Het verstrekken van cijfers hoeft dan niet een eenmalige zaak te zijn, zonder dat er enig inzicht bestaat in de konklusies waartoe de cijfers leiden. In de volgende hoofdstukken behandelen we de methoden die ook voor de automatiseerders zijn beschreven. Nu wordt echter de kant van de gebruikers besproken.

Kommunikatie betekent in dit verband voor de gebruikers: eisen stellen en gegevens verstrekken, maar nog tijdens het logisch ontwerp de konsekwenties overzien van de gestelde eisen en verstrekte gegevens.

Hoewel het aksent ligt op ontwerpmethoden, moet er toch nog iets gezegd worden over de analysefase. In die fase brengen informatie-analisten en gebruikers de bestaande situatie in kaart. Een probleem voor de informatie-analisten is de mate van gedetailleerdheid. Als er orders worden ingevoerd zou de afhandeling door

de gebruiker af kunnen hangen van de bestelde artikelen. Er bestaat geen verschil in de administratieve verwerking, maar de gebruiker moet bij bepaalde orders bijvoorbeeld de offerte erbij nemen en de bestelling kontroleren aan de hand van de levertijd. De gebruiker kan tijdens de analyse de informatie-analist vertellen dat er 1000 orders per dag verwerkt worden. Hij kan ook wijzen op het verschil tussen standaard orders en orders waar een offerte bijhoort. Kortom, het kan geen kwaad iets meer te vertellen dan er gevraagd wordt. Een uitgebreide toelichting op de struktuur, de stijl en het soort papier van de offerte is minder zinvol voor het te ontwerpen computersysteem. De gebruiker moet dus aan de ene kant proberen zoveel mogelijk informatie te verstrekken en aan de andere kant moet hij aksepteren dat de informatie-analist soms zijn verhaal afbreekt, omdat het geen informatie bevat waar hij iets aan heeft. Hetzelfde geldt natuurlijk voor de informatieverstrekking door de informatie-analist.
Voor de gebruiker moet het kernpunt blijven: het gaat om de analysefase, de ontwerpfase komt nog. Tijdens het ontwerp mogen kreatieve gebruikers meer vragen dan informatie-analisten hebben bedacht op basis van de analyse.
Eerst meer vertellen dan wordt gevraagd, later meer vragen dan wordt verteld.

61.6 Zelfdiscipline

In een gezin wordt de warme maaltijd voor de volgende dag besproken. Gezien de prijzen en het seizoen wordt gekozen voor spruitjes met aardappelen en vlees. De volgende dag, nadat de inkopen zijn gedaan, komt een van de gezinsleden op het idee dat een stoofschotel van puree, kaas en spruitjes eigenlijk veel leuker is. De kok kijkt of er nog kaas is en als dat het geval blijkt te zijn gaat het vlees de diepvries is en wordt de kaas geraspt. Als de schotel in de oven staat komt pa van z'n werk. Hij is het hele gesprek van de vorige avond vergeten en heeft alleen trek in stamppot van boerenkool met worst. Hem wordt liefdevol duidelijk gemaakt dat dat wel kan, maar dan morgenavond. Tijdens het logisch ontwerp van de maaltijd stonden binnen het budget en beschikbaarheid van de groente, alle mogelijkheden nog open. Tijdens het technisch ontwerp bleek er nog te kunnen worden overgeschakeld naar een stoofschotel omdat er toevallig kaas in huis was. Toen pa tijdens de bouw van de maaltijd nog over wilde schakelen op een andere groente en ander vlees bleek dat niet te kunnen. Veel gebruikers die al enige ervaring hebben in het werken met computer en iets begrepen denken te hebben van de mondigheid van gebruikers ten opzichte van automatiseerders, lijken op pa. Soms gaan ze zover dat ze, onafhankelijk van wat er al gedaan is,

toch stamppot van boerenkool eisen. Dat betekent dat er alsnog boerenkool en worst moet worden aangeschaft, dat de stoofschotel kan worden weggegooid en dat die avond de vaat samenvalt met het late journaal. Uiteindelijk is niemand tevreden. De automatiseerders niet omdat ze opnieuw moesten beginnen, de gebruikers niet omdat het geheel niet op tijd klaar was en veel meer kostte dan was voorzien.
Het logisch ontwerp is de fase waarin gebruikers kunnen aangeven wat ze willen. Hoelang die fase duurt, hangt van het projekt af. Soms begint men voor het ene gedeelte van de toepassing reeds aan het technisch ontwerp terwijl andere delen zich nog bevinden in de fase logisch ontwerp. Gebruikers moeten dus zorgen dat ze weten wanneer het deel dat hen betreft overgaat naar de fase technisch ontwerp. Als het technisch ontwerp is begonnen is de tijd van inspraak voorbij.
Natuurlijk is het zo dat een bedrijf een dynamisch geheel is. Alles is altijd in beweging en in ontwikkeling. De automatisering zal moeten volgen, maar wel met een tempo dat past bij de tijd die nodig is om systemen te bouwen en aan te passen. Grote, kostbare aanpassingen gebeuren niet iedere dag. Het veranderen van de organisatie of de huisvesting zal vaak nodig zijn, maar dat doet men meestal niet iedere maand. Planningen ten aanzien van deze ingrepen beslaan meestal een jaar of langer. Bovendien heeft niet iedere verandering in het personeelsbestand direkt een andere huisvesting tot gevolg. Grote veranderingen gaan sprongsgewijs maar hebben een lage frekwentie. Voor aanpassingen van het geautomatiseerd systeem geldt hetzelfde. Bij nieuwbouw ontstaat de eerste versie van de software. Na enige tijd maken allerlei wijzigingsvoorstellen het noodzakelijk een tweede versie te maken. Ook die voorstellen doorlopen weer de fasen van de projektaanpak. Bij nieuwbouw is het einde van het logisch ontwerp tevens het einde van gebruiksinbreng. Alles wat daarna nog aan voorstellen binnenkomt bij de automatiseerders wordt bewaard voor de tweede versie van de software. Daar hebben de gebruikers zelfdiscipline voor nodig. In de praktijk komt het regelmatig voor dat de automatiseerders vele malen opnieuw kunnen beginnen aan het technisch ontwerp. Veel projekten lopen een enorme vertraging op doordat gebruikers tijdens de bouw nog met nieuwe plannen komen. Als we de verslagen van de Centrale Rekenkamer mogen geloven over projekten bij de overheid, die helemaal niet van de grond komen, dan zou dat weleens veroorzaakt kunnen zijn door gebrek aan deskundigheid van de automatiseerders, maar evengoed door gebruikers die geen uitspraken doen of door de slechte kommunikatie tussen beide groepen. Grondprobleem bij gebruikers is vaak, dat ze geen volledig beeld hebben van wat automatisering nu precies per werkplek betekent en dat ze niet kunnen overzien wat de gevolgen zijn

van uitspraken die ze doen, kwalitatief noch kwantitatief. Bij de behandeling van methoden voor het ontwerpen van interaktieve toepassingen zal blijken dat tijdens het logisch ontwerp de noodzakelijke terugkoppeling naar de gebruiker in voldoende mate is te realiseren. In sommige gevallen kunnen de definitieve, gedetailleerde konsekwenties pas tijdens het technisch ontwerp worden aangegeven. Wanneer dat nog zou leiden tot kleine aanpassingen van het ontwerp, kost dat misschien wat tijd en geld, maar dat is normaal in het komplexe gebeuren van automatisering. Dat is dan nog altijd heel iets anders dan projekten die een veelvoud kosten van het oorspronkelijke budget of zelfs helemaal niet van de grond komen.

Hoofdstuk 62
Transakties

62.1 Wat zijn transakties

Het begrip transakties is algemeen bekend. Een verkoop wordt in sommige branches een transaktie genoemd. Automatiseerders gebruikers in hun vakjargon soms het woord transactions en bedoelen dan iets wat in het inwendige van een computer plaatsvindt. Wij zullen het begrip transaktie als volgt weergeven: de procedure aan een beeldscherm, zoals die door de gebruiker wordt ervaren. Op die ervaring komen we later terug bij ingewikkelde transakties. Een belangrijk aspekt van die procedure aan het beeldscherm is dat de gebruiker kan aangeven hoevaak hij die per dag uitvoert. Het is een soort eenheid van werk. In de handmatige situatie kennen we talloze soorten procedures: het aksepteren van orders, het maken van kontrakten, het kontroleren van de voorraad, het doen van boekingen enzovoort. Wanneer zo'n procedure geautomatiseerd wordt met een beeldscherm spreken we van een transaktie. Een transaktie is dus de geautomatiseerde versie van een procedure. Natuurlijk kunnen er transakties ontworpen worden om via het beeldscherm dingen te doen, die in de handmatige situatie helemaal niet mogelijk zijn. Soms wordt van een handmatige procedure maar een heel klein deel geautomatiseerd. Dan bestaat de transaktie maar voor een klein deel uit werken met het beeldscherm. Het is ook mogelijk dat er binnen een procedure twee keer even iets

met het beeldscherm wordt gedaan, maar niet altijd en niet altijd twee keer. In zo'n geval zal de informatie-analist voorstellen twee transakties te ontwerpen die naar keuze kunnen worden uitgevoerd. Als de gebruiker goed in de gaten houdt hoe hij van de ene naar de andere transaktie over moet schakelen en er als mede-ontwerper voor zorgt dat dat gebruikersvriendelijk gaat, is voor hem het vaststellen van twee transakties geen probleem.
De vertaling van procedures naar transakties hoeft dus niet altijd één op één plaats te vinden. De gebruiker moet zich realiseren dat per transaktie gevraagd zal worden naar de frekwentie ervan. Dat kan het aantal keren per dag zijn, dat de transakties worden uitgevoerd of per week of per maand en hoe de pieksituaties er uit zien.
Onder eenvoudige transakties verstaan we transakties die uit een reeks handelingen bestaan die altijd in dezelfde volgorde worden uitgevoerd. De transaktie afgebeeld in Fig. 61.1 is een eenvoudige transaktie. Daarbij is het niet belangrijk om hoeveel schermlay-outs het gaat. Het zou kunnen zijn dat er één schermlay-out is ontworpen die tijdens de transaktie wordt gevuld door ingetypte en gedisplayde gegevens. In zo'n geval worden aan het eind van de transaktie de genoemde gegevens vervangen door spaties en de volgende transaktie kan beginnen. Anders gezegd: het scherm wordt schoongemaakt, maar het masker, de schermlay-out, blijft staan.
Het zou echter ook kunnen zijn dat per interaktie een nieuwe schermlay-out verschijnt.
In Fig. 62.1 is de situatie iets ingewikkelder. Het tweede deel van de transaktie FAKTUURKONTROLE bestaat in 70% van de gevallen uit het aksepteren van een korrekte faktuur, in 30% van de gevallen moeten een of meer bedragen worden aangepast. In principe zou hier al de vraag gesteld kunnen worden of er twee transakties bestaan: KORREKTIE FAKTUREN en FOUTE FAKTUREN? In het voorbeeld van Fig. 62.1 is de vraag niet erg belangrijk, omdat de transaktie FAKTUURKONTROLE overzichtelijk genoeg is. Het enige verschil is dan de vraag van de informatie-analist naar de kwantiteiten. Bij FAKTUURKONTROLE zijn de vragen:
- Hoeveel fakturen worden per dag gekontroleerd?
- Hoeveel procent daarvan is goed?
- Hoeveel procent daarvan is fout?
De laatste vraag is in dit simpele voorbeeld natuurlijk overbodig, maar in een ingewikkelde situatie is het antwoord nuttig ter kontrole. Bij een splitsing in twee transakties worden de vragen:
- Hoeveel korrekte fakturen worden per dag verwerkt?
- Hoeveel foute fakturen worden per dag afgehandeld?
Als we de verwerking door de computer even buiten beschouwing laten, kunnen we Fig. 62.1 ook weergeven als Fig. 62.2.
We zullen de situatie nog iets ingewikkelder maken door aan te

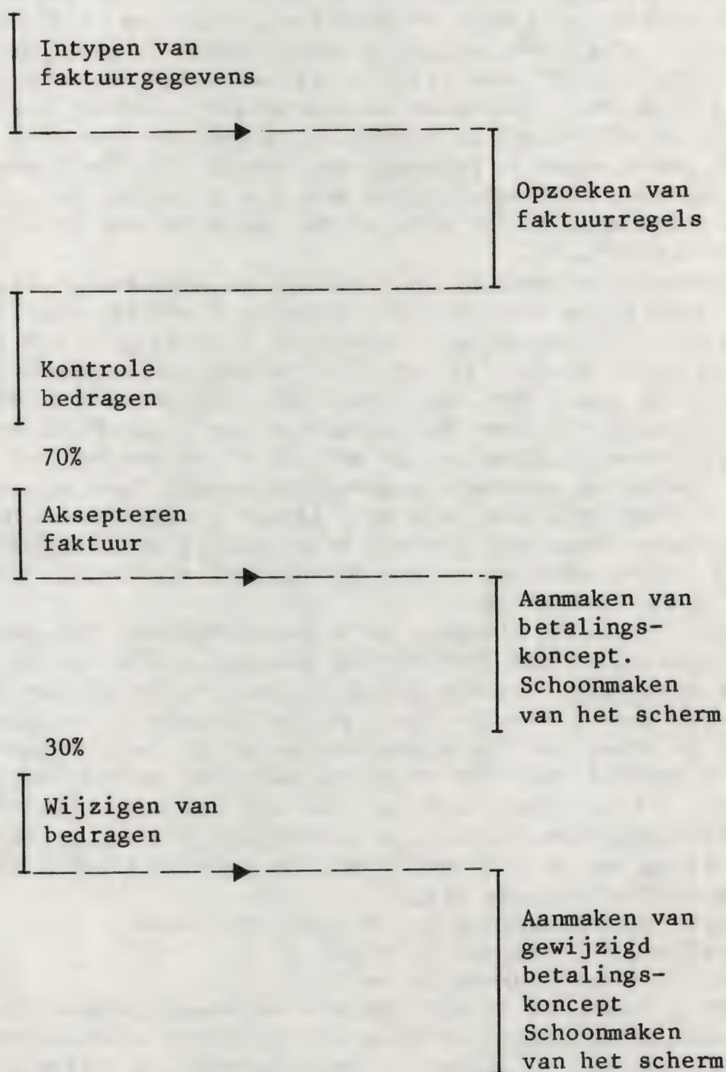


Fig. 62.1 Voorbeeld van een transaktie

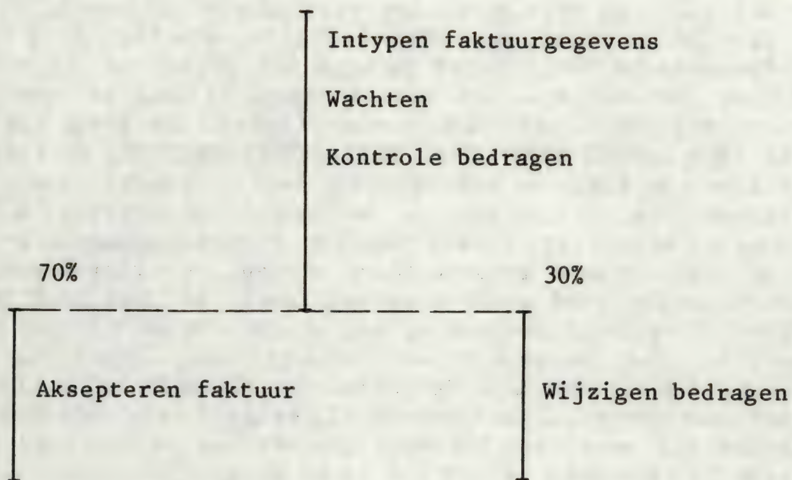


Fig 62.2 Twee situaties in een transaktie

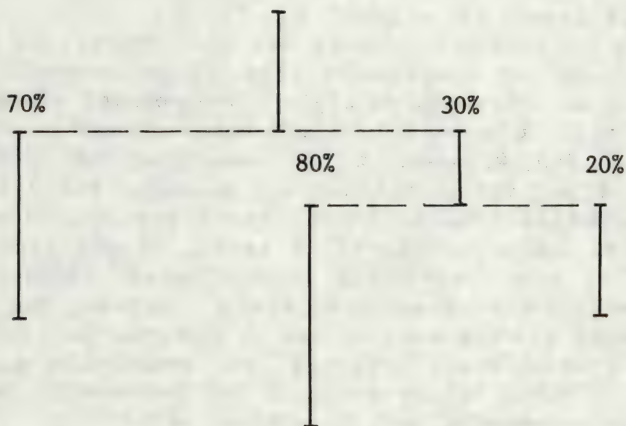


Fig. 62.3 Drie situaties in een transaktie.

nemen dat fakturen onjuist kunnen zijn doordat de aantallen niet juist zijn of de bedragen niet juist zijn, zie Fig. 62.3 Van de foute fakturen is 80% onjuist vanwege het aantal en 20% vanwege het bedrag. Dat betekent dat weer de vraag gesteld kan worden of de transaktie FOUTE FAKTUREN in twee transakties gesplitst moet worden: FOUT BEDRAG FAKTUUR en FOUT AANTAL FAKTUUR. Stel dat de afhandeling van fakturen met een fout aantal erop een geheel andere afhandeling vereist dan die met een onjuist bedrag. Bij een fout aantal moet bijvoorbeeld kontakt worden opgenomen met het magazijn, de voorraad gekontroleerd worden, een kontrolebon van een handtekening worden voorzien enzovoort. Bij een fout bedrag mag de prijs veranderd worden op het scherm. In zo'n situatie zal de gebruiker heel duidelijk twee verschillende transakties herkennen. Misschien werden in de handmatige situatie beide transakties wel door verschillende mensen afgehandeld. Als de gebruikers ze ervaren als twee verschillende transakties is het goed om de splitsing in te voeren. Eigenlijk is er nauwelijks sprake van een splitsing. De gebruiker zou immers de samenvoeging tot een transaktie niet als logisch ervaren. De informatie-analist komt meestal van de andere kant en brengt de afhandeling van fakturen in kaart. Voor hem is het een splitsing. In het algemeen zijn automatiseerders geneigd om dingen samen te voegen, omdat ze op elkaar lijken wat de verwerking door de computer betreft. Soms is er qua verwerking bijna geen verschil tussen het invoeren van een order en het wijzigen van een order. Voor de gebruiker kan het over twee verschillende taken gaan.

De ervaring van de gebruiker speelt dus een belangrijke rol bij het vaststellen van wat transakties zijn. Of de automatiseerders voor hun technische rekenwerk de cijfers van sommige transakties toch weer kombineren tot een geheel, is niet belangrijk.

In de paragraaf Projektaanpak is al aangegeven dat gebruikers mede-ontwerper zijn tijdens het logisch ontwerp. Pas dan worden transakties ontworpen. Daaraan voorafgaand brengen informatie-analisten de bestaande procedures in kaart. Of dat globaal gebeurt tijdens het vooronderzoek of gedetailleerd tijdens de analysefase is voor gebruikers niet van belang. De vraag die aan informatie-analisten gesteld moet worden is die over de nauwkeurigheid van de te verstrekken gegevens. Hoe nauwkeurig moeten de percentages zijn? Wordt er ook gevraagd naar het aantal regels per faktuur? Hoe nauwkeurig moet dat aantal zijn?

Het is duidelijk dat het voor de gebruiker van weinig belang is, op welk moment in de projektaanpak de gegevens gevraagd worden. De mate van gedetailleerdheid en de gewenste nauwkeurigheid bepalen de hoeveelheid werk en de planning ervan.

Het is belangrijk dat de gebruikers in de gaten houden of het gaat om analyse of om ontwerp. Zolang het om analyse gaat hoeven

gebruikers zich geen zorgen te maken, als voor het ontwerp de toepassing van methoden is overeengekomen.

62.2 Het ontwerpen van transakties

Wanneer de bestaande situatie door de informatie-analisten in samenwerking met de gebruikers in kaart is gebracht, kunnen transakties worden ontworpen. Aansluitend bij het voorbeeld in de vorige paragraaf kunnen we vaststellen dat er fakturen worden gekontroleerd. Als tijdens de analyse alleen is vastgesteld dat er een procedure is om fakturen te kontroleren, dan zouden tijdens het logisch ontwerp een gebruiker en een informatie-analist kunnen beginnen met het ontwerpen van de transaktie FAKTUURKONTROLE. Dan zou al gauw blijken dat bij de foute fakturen twee heel verschillende procedures worden uitgevoerd. Misschien is voor het afhandelen van onjuiste aantallen wel een andere gebruiker nodig. Die transaktie moet later worden ontworpen met die andere gebruiker. De analyse is dan te onnauwkeurig geweest. In het deel voor de informatie-analisten is dan ook aangegeven dat analyse zo gedetailleerd moet zijn dat er uitzicht ontstaat op een beeldschermversie van de procedure, een transaktie. Een waterdichte grens voor de detaillering is niet aan te geven. Als de informatie-analist pas tijdens het logisch ontwerp tot de konklusie komt dat er meer transakties ontstaan dan vastgestelde procedures, is er voor de gebruiker niets aan de hand, omdat op dit moment pas de transakties worden ontworpen. Als de analyse te onnauwkeurig is geweest en er een planning op is gebaseerd, kan er wel iets fout gaan.
Hoe meer ervaring de informatie-analist heeft of hoe meer kennis hij van de materie heeft, des te kleiner de kans op dit soort oneffenheden. Het is van belang dat gebruikers tijdens de analyse waar mogelijk dit soort problemen voorkomen door iets meer te vertellen dan gevraagd wordt. In ieder geval vormen de in kaart gebrachte procedures de basis voor de transakties. Daarnaast kunnen transakties worden bedacht en voorgesteld door kreatieve gebruikers.
Ontwerpen is een iteratief proces. Er worden in eerste instantie eisen gesteld door transakties te ontwerpen. De automatiseerders kunnen door toepassing van bepaalde rekenmethoden vaststellen wat de konsekwenties zijn voor de apparatuur, de benodigde tijd om die transakties te realiseren op de computer. Als de gebruiker wordt gekonfronteerd met de konsekwenties, kan het zijn dat hij een aantal eisen of transakties laat vervallen. Opnieuw rekenen de automatiseerders de situatie door en komen met nieuwe cijfers. Dit kan zich enige keren herhalen. Dit iteratieve proces is alleen mogelijk wanneer

- er volgens afgesproken methoden wordt ontworpen,
- er een vertaling mogelijk is van eisen naar konsekwenties.

Als die methoden worden toegepast ontstaat er zowel voor gebruikers als voor automatiseerder, een stuk duidelijkheid over de realisering van kreatieve gedachten.
Een voorwaarde bij dit proces is, dat de toekomstige situatie al tijdens het ontwerp ervaren wordt door gebruikers. Vooral gebruikers die nog nooit met een beeldscherm hebben gewerkt, kunnen zich bepaalde transakties helemaal niet voorstellen, laat staan dat ze kunnen beoordelen of het realiseren ervan opweegt tegen de kosten. Het iteratieve proces kan zich afspelen binnen het logisch ontwerp. Het einde van het logisch ontwerp is het einde van het ontwerpproces van de gebruikers.
Tijdens de fase technisch ontwerp gaan de systeemontwerpers proberen het logisch ontwerp te vertalen naar een technisch ontwerp. Gebruikers moeten op een of andere manier het eind van het technisch ontwerp in de gaten houden. Dan moeten ze de automatiseerders laten aangeven of aan alle gestelde eisen zal worden voldaan. Verstandige ontwerpers die weten dat dat niet gaat, geven al eerder een signaal dat iets niet kan. Dan ontstaat er soms een iteratie vanuit het technische ontwerp naar het logisch ontwerp. Als een transaktie namelijk volledig moet worden herzien, moet voor die transaktie het ontwerpproces herhaald worden en dat begint bij het logisch ontwerp. In veel bedrijven worden geen methoden toegepast om gebruikers te konfronteren met de konsekwenties van hun eisen, zo ze die al mochten stellen. Daar wordt het hele systeem eerst gebouwd en dan ziet de gebruiker pas, hoe de geautomatiseerde situatie er uit ziet. Natuurlijk bestaat er voor gerealiseerde en gevoerde systemen nog zoiets als een evaluatie. Dan kan het systeem dus ook nog aangepast worden aan de eisen van de gebruikers. Men zou dat ook nog een iteratie kunnen noemen.
Iteraties tijdens het logisch ontwerp duren dagen, iteraties via het technisch ontwerp weken tot maanden en die via de invoering van het systeem jaren! Tot nu toe hebben we steeds gesproken over methoden die moeten worden toegepast. In de volgende hoofdstukken zullen we die methoden bespreken. Het gaat daarbij om de aspekten voor de gebruikers; die voor de automatiseerders zijn behandeld in de delen voor de automatiseerders. Alles draait om het ontwerpen van transakties. Dat ontwerpen houdt in dat de gebruikers mede-ontwerper zijn van de procedure aan het beeldscherm, die procedure kunnen uitvoeren en evalueren en dat de automatiseerders de gevolgen van het ontwerp voor de gebruikers voor het systeem kunnen berekenen.
Bij dat transaktie-ontwerp gaat het om twee methoden: dialoogsi-dialoogsimulatie en Transaktie analyse, zie Fig. 62.4.
Dialoogsimulatie is een methode waarbij de procedure in gebrui-

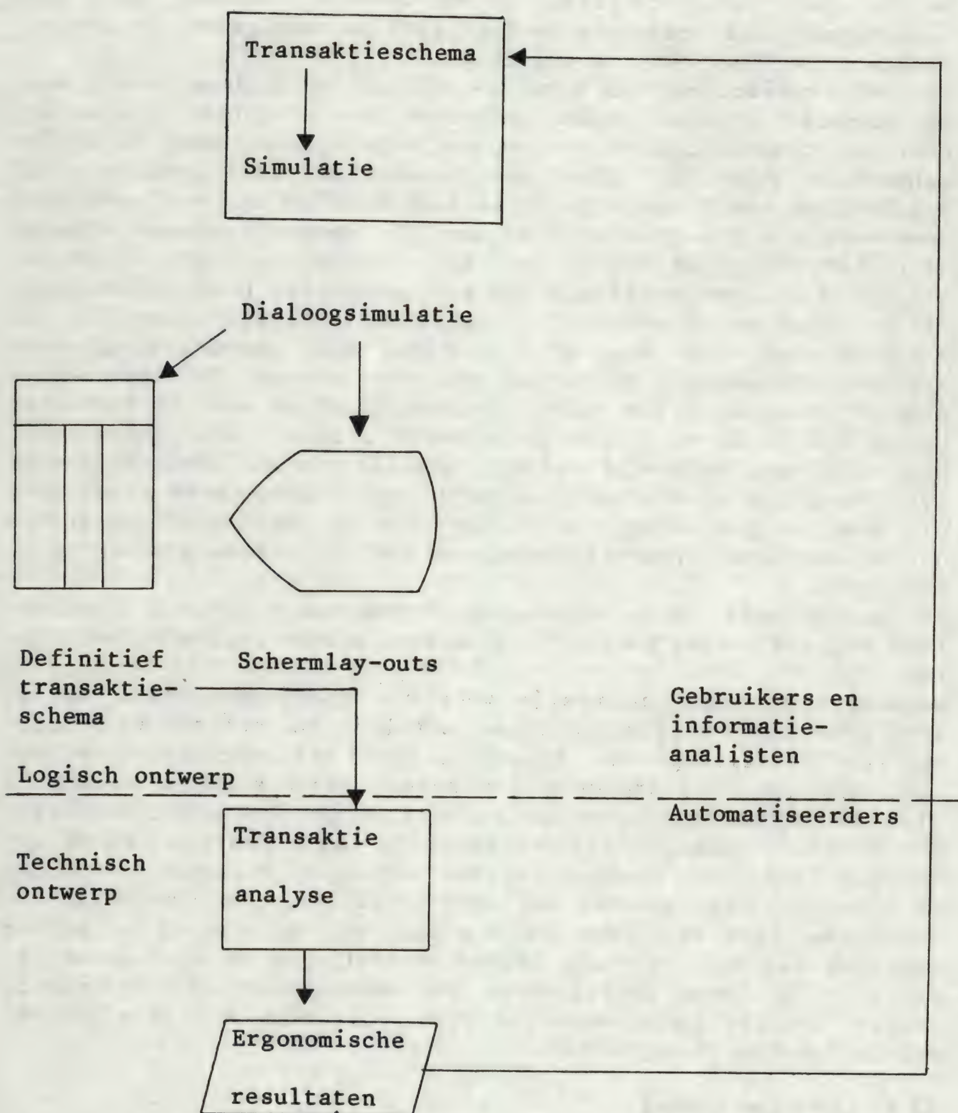


Fig. 62.4 Transaktie-ontwerp

kerstermen wordt vastgelegd en vervolgens "live" wordt uitgevoerd op een beeldscherm.
De procedure wordt vastgelegd op een transaktieschema. Een transaktieschema ziet eruit als in Fig. 62.5 is aangegeven. De overeenkomst met Fig. 61.1 is duidelijk.
Als het transaktieschema klaar is, is ook de dialoog tussen mens en computer bepaald. Omdat ontwerpen een iteratief proces is, moet het transaktieschema eerst een voorlopig dokument zijn. Pas wanneer de gebruiker achter het beeldscherm heeft gezeten en 20 transakties heeft uitgevoerd, is bekend of er nog wat veranderd moet worden aan de procedure en aan het transaktieschema. Wanneer alle betrokken gebruikers met het beeldscherm hebben gewerkt blijkt of we kunnen spreken van een definitief transaktieschema. Het op deze wijze definitief geworden transaktieschema is het startdokument voor Transaktie analyse. Deze methode zorgt voor een kwantificering bij interaktieve toepassingen. Een transaktieanalist vertaalt alles op het transaktieschema naar cijfers. Een belangrijk deel van die cijfers wordt geleverd door gebruikers, andere cijfers hangen af van het computersysteem. Transaktie analyse levert twee soorten resultaten op: ergonomische resultaten die voor de gebruikers van belang zijn en technische resultaten die de automatiseerders gebruiken om het technische systeem te ontwerpen.
Zo kan op basis van de ergonomische resultaten van alle transakties bepaald worden hoeveel uren mensen achter beeldschermen zitten.
Nogmaals, de resultaten zijn voor een deel afhankelijk van de door gebruikers verstrekte cijfers. Ook in dat opzicht is ontwerpen een iteratief proces. Sommige cijfers zijn moeilijk te schatten. Als echter gebruikers een aantal cijfers mogen noemen en zien tot welke konklusies de cijfers leiden dan wordt duidelijk hoe kritisch bepaalde cijfers zijn. Op basis daarvan kan de gebruiker besluiten sommige cijfers nog eens nauwkeurig vast te stellen. Als hij, wanneer dat niet mogelijk is, een bepaalde reservekapaciteit wil inbouwen dan weet hij nu precies de konsekwenties daarvan. Kortom, dialoogsimulatie is de methode om gebruikers te laten funktioneren als mede-ontwerpers, Transaktie analyse levert, zoals we zullen zien, de sociale aspekten van de automatisering in cijfers.

62.3 Cijfers gevraagd

Naast de enorme hoeveelheid informatie die automatiseerders nodig schijnen te hebben, willen ze ook nog cijfers hebben. Soms cijfers waarover geen enkele gebruiker ooit heeft nagedacht.
Als het goed is wordt bij elke bedrijfsinvestering onderzocht wat

Transaktieschema centraal

Transaktienaam: BEHEER VERKOOPPRIJZEN

Menselijke handelingen	Transport	Machinale verwerking
Intoetsen van artikel-nummer of -groep en vestigingsnummer		
	-----)	Raadplegen: Artikelbestand, Historische gegevens Opbouw voor displayen
	(-----	
Intoetsen van de parameters	-----)	Kontrole + berekening model oud + model nieuw. Displayen
	(-----	
Kontrole + indrukken F-toets voor hard copy		
	-----)	Voorbereiding voor printer, terug naar menuscherm.

Fig. 62.5 Voorbeeld van een transaktieschema.

de kosten en de baten zijn en op welke termijn. Nu is een investering in een typemachine er een van een andere orde dan die in een nieuw gebouw. De kosten van de automatisering benaderen in het algemeen eerder die van een gebouw dan die van een typemachine. Een dergelijke kosten/baten-analyse is dus redelijk. Ook al zouden de baten moeilijk te kwantificeren zijn, dat kan geen reden zijn om de kosten niet zo nauwkeurig mogelijk in kaart te brengen.
Als het klantenbestand in het geheugen van de computer moet worden ondergebracht dan is het zelfs de meest a-technische gebruiker duidelijk, dat de grootte van dat geheugen afhangt van het aantal klanten. Als de grootte van het geheugen de prijs van de computer bepaalt dan is die gebruiker ook duidelijk dat de vraag naar het aantal klanten gerechtvaardigd is. Met een methode als Transaktie analyse kan onder andere het aantal benodigde beeldschermen worden berekend. Het aantal beeldschermen heeft uiteraard invloed op de kosten van het hele systeem en het is minstens zo belangrijk voor de gebruiker zelf. Het voordeel van Transaktie analyse is dat altijd precies kan worden aangegeven welke cijfers tot bepaalde konklusies hebben geleid en hoe.
Het opstellen van transaktieschema's door informatie-analisten en gebruikers samen maakt het mogelijk gebruikers een goed inzicht te geven in de cijfers die gevraagd worden. Als er op het transaktieschema staat dat de computer orderregels moet displayen, komt zeker de vraag om hoeveel regels het gaat. Als er op het transaktieschema staat dat er soms een andere procedure moet worden gevolgd, dan wordt een keer gevraagd in hoeveel procent van de gevallen dat voorkomt. Op het transaktieschema mogen best algemene aanduidingen blijven staan, maar de gebruiker heeft nu een duidelijke richtlijn voor de cijfers die hij nog moet verzamelen. Het kan geen kwaad om van sommige cijfers te vragen waarom ze gevraagd worden. Het verzamelen van cijfers kost vaak tijd: soms gaat het immers om cijfers waar gebruikers nog nooit over nagedacht hebben. De benodigde tijd mag best gepland worden. Daarom is het noodzakelijk dat informatie-analisten aangeven om welke cijfers het gaat en wat de gewenste nauwkeurigheid is.
Laten we als voorbeeld nemen de orders die op een verkoopafdeling worden verwerkt.
Het aantal orders per dag is een getal dat niet iedereen precies weet, maar iedere verkoper heeft er wel een idee van. Het aantal orderregels per order is vaak al veel moeilijker aan te geven. Over het aantal letters in een artikelomschrijving heeft waarschijnlijk nog nooit iemand nagedacht.
De informatie-analist moet in de vorm van een vragenlijst aangeven welke cijfers hij nodig heeft en hoe nauwkeurig die moeten zijn. Die nauwkeurigheid zal afhangen van het soort onderzoek.

Tijdens het vooronderzoek gaat het om schattingen over langere periodes, tijdens het logisch ontwerp gaat het om nauwkeurigheden van 10% bij kritische cijfers.
Op basis van deze gegevens moet de gebruiker een planning opstellen om de gewenste cijfers te bepalen. Meestal gunt niemand zich de tijd om ze te verzamelen en wordt er gewerkt met ruwe schattingen. Transaktie analyse levert wel direkt de konklusies van ruwe schattingen. Ruwe aantallen beeldschermen zijn voor de meeste gebruikers niet akseptabel, evenmin als ruwe schattingen van het aantal uren dat per dag met een beeldscherm moet worden gewerkt. Cijfers moeten in het belang van de gebruikers zelf zo nauwkeurig mogelijk zijn. Een groot aantal cijfers is per dag of per situatie verschillend. Het aantal orders per dag en het aantal posities op een order zullen geen vaste getallen zijn.
Transaktie analyse biedt de mogelijkheid de spreiding in die cijfers mee te nemen in de berekeningen. De cijfers kunnen worden aangeboden in de vorm van een gemiddelde waarde en een standaardafwijking of een variantie. Gebruikers kunnen deze cijfers zelf bepalen of een tabel aan de informatie-analist overhandigen met aantallen orders per dag die gedurende een bepaalde periode zijn gemeten. Bij nieuwe transakties waarvan nog geen ervaringscijfers bestaan kunnen ze bijvoorbeeld minima en maxima opgeven. Direkt nadat Transakties analyse is uitgevoerd kan bekeken worden wat de konsekwenties zijn van de schattingen.
Als de gebruiker niet het gemiddelde en de standaardafwijking kan aangeven, moet hij in ieder geval aangeven wat het gemiddelde is, met een onder- en een bovengrens.
Soms kunnen cijfers aanleiding zijn een procedure te splitsen in twee transakties. Stel dat in het geval van het aantal orderregels per order, de gebruiker niet kan spreken van een gemiddeld aantal en een standaard afwijking omdat er soms bulkorders voorkomen van een paar honderd regels. Dat kan reden zijn om twee transaktie vast te stellen; ORDERS en BULKORDERS. Het transaktieschema is van beide hetzelfde, maar de transaktie ORDERS levert via Transaktie analyse heel andere resultaten op, dan die van BULKORDERS. Zo kunnen per geval oplossingen bedacht worden om ogenschijnlijk onduidelijke situaties toch redelijk in kaart te brengen.
In het voorbeeld moet de gebruiker nu natuurlijk wel aangeven hoeveel orders en bulkorders hij per dag verwerkt en hoeveel orderregels op een order en een bulkorder voorkomen.
Via Transaktie analyse leiden cijfers tot cijfers. Cijfers van gebruikers leveren cijfers op over de geautomatiseerde situatie. Die vertaling maakt het leveren van cijfers gemakkelijker. De gebruiker kan de zaak door laten rekenen voor verschillende cijfers en zo zijn schattingen afwegen tegen de konsekwenties die

ze hebben.
De vertaling geeft ook inzicht in allerlei uitzonderingssituaties. Als het aantal beeldschermen is berekend op basis van het gemiddelde aantal orders, hoeveel orders moeten dan tot de volgende dag blijven liggen in een pieksituatie?
Het effekt van cijfers kan voor gebruikers positief en negatief zijn. Verhalen over werkeloosheid ten gevolge van automatisering blinken meestal uit in algemeenheden. Als het aantal benodigde beeldschermen op een afdeling kleiner is dan het aantal funktionarissen, dan is dat bekend tijdens het logisch ontwerp. In veel bedrijven is men helemaal niet van plan mensen te ontslaan, maar wil men effektiever gaan werken door andere aktiviteiten op te pakken. Als het aantal beeldschermen groter is dan het aantal funktionarissen dan betekent dat dat er meer mensen nodig zijn. In ieder geval geldt dat de berekeningen op basis van de ergonomische resultaten van Transaktie analyse zijn gebaseerd op het transaktie-ontwerp en de cijfers van de gebruikers. De konklusies zijn nu bekend tijdens het logisch ontwerp en dan kan er nog heel wat veranderd worden!
Bedrijfspolitieke problemen zullen door geen enkele methode worden opgelost: wanneer een gebruiker merkt dat zijn situatie kwantitatief in kaart wordt gebracht, kan hij dat terecht of ten onrechte als een bedreiging beschouwen en op grond daarvan elke medewerking aan het verstrekken van cijfers weigeren of zelfs onjuiste cijfers verstrekken. Het is voor informatie-analisten niet moeilijk aannames te doen en op basis daarvan hun berekeningen uit te voeren. De gebruikersorganisatie mag de resultaten van de berekeningen kontroleren of direkt beginnen met de verifikatie van de aannames. Daarmee ligt het probleem bij de gebruikers en daar hoort het ook. Zo kunnen cijfers soms aanleiding zijn om sociale aspekten van de automatisering boven water te krijgen. Sociale problemen moeten echter niet opgelost worden door automatiseerders: personeelszaken is een onderdeel van de gebruikersorganisatie.

Hoofdstuk 63
Dialoogsimulatie

63.1 Dialoogsimulatie als methode

Bij interaktieve toepassingen werken gebruikers met een beeldscherm. Ze voeren transakties uit die voor een deel bestaan uit de dialoog met de computer via T.V-scherm en toetsenbord. De gebruiker typt iets in, de computer reageert en zet na enkele seconden iets op het scherm. Wat de computer in die seconden precies doet is voor de gebuiker op dat moment niet van belang. Dat is een keer vastgesteld en vastgelegd in de vorm van programma's. Het gaat om de dialoog en de tijd die de gebruiker moet wachten op de computer. Met een dialoogsimulator kan dat proces worden gesimuleerd. De schermlay-out wordt op het scherm van de simulator ontworpen en werkt meteen, alsof alle programma's al gemaakt waren. De simulator is een kleine portable microcomputer die kan worden aangeschaft zonder dat er nog is nagedacht over de eigenlijke computer. Dat betekent dat er tijdens het logisch ontwerp al op iedere werkplek met het beeldscherm kan worden gewerkt zonder dat het uiteindelijke computersysteem aanwezig hoeft te zijn. Tijdens het logisch ontwerp zien en ervaren de gebruikers precies wat er gebeurt en tijdens de rest van de looptijd van het projekt kunnen ze funktionele en organisatorische wijzigingen al voorbereiden.
Een methode is een uitgekristalliseerde, gestandaardiseerde en

gedokumenteerde manier van werken. Goede methoden hebben hun nut bewezen. Dialoogsimulatie is een methode om gebruikers en informatie-analisten samen een systeem te laten ontwerpen. Ieder werkt daarbij van uit zijn eigen denk- en belevingswereld. Dialoogsimulatie is dus voor een gebruiker iets anders dan voor een informatie-analist. Daarom is de voorbereiding van beide ook geheel verschillend (15, 16). Wanneer beiden goed zijn voorbereid, weten ze precies hoe er gewerkt moet worden en wat er moet gebeuren. Het is van belang dat gebruikers zich niet laten verleiden tot een andere manier van ontwerpen. Van andere methoden zijn de resultaten niet meer kontroleerbaar. Het verschil in vakgebied is zo groot, dat het voor een gebruiker niet te beoordelen is wat de gevolgen zijn van een andere aanpak. Goede methoden hebben in de praktijk hun nut bewezen en de werkwijze is vastgelegd. Dialoogsimulatie is het eerste onderdeel van het transaktie-ontwerp en bestaat uit de volgende stappen:

- het maken van transaktieschema's
- het maken van een startontwerp op de simulator
- de evaluatie van het ontwerp
- de simulatie, eventueel verscheidene malen voor verschillende groepen
- de evaluatie en het maken van het definitieve transaktieschema
- de overdracht van ontwerpdokumenten aan de gebruikers.

We zullen nu per stap een aantal opmerkingen maken.

- Het maken van transaktieschema's.

Voor veel automatiseerders is dat een moeilijke stap, omdat ze gewend zijn schermlay-outs als eerste en laatste gebruikersdokument te maken. Bovendien kunnen op de simulator heel gemakkelijk schermlay-outs worden gemaakt en aangepast. Hoe eerder ze weer op het automatiseringsterrein zijn hoe liever. Gedurende de lange maanden van het ontwerp en de bouw heeft de gebruiker dan niets anders dan mappen met schermlay-outs waaruit de nieuwe manier van werken niet of nauwelijks is af te leiden. En dan zijn automatiseerders nog verbaasd dat gebruikers zo moeilijk doen over wat uitstel van de opleveringsdatum.

Transaktieschema's moeten worden gemaakt. Ze geven de gebruiker een duidelijk beeld van de procedure, de aansluiting op de handmatige procedures en verandering in de hele werkwijze. Op basis van transaktieschema's kan al een begin gemaakt worden met de gebruikershandleiding tijdens het technisch ontwerp. Ten gevolge van tijdens het technisch ontwerp gekonstateerde problemen, is misschien wel eens een aanpassing nodig, maar die betreffen meestal maar enkele transakties. Wil men dat risico helemaal uitsluiten dan kan men wachten tot de bouw is begonnen.

Transaktieschema's zijn voor de gebruikers de akseptatiedokumenten tijdens de oplevering. Ze vormen het begin van nieuwe taak/-funktie-omschrijvingen. Personeelszaken en ondernemingsraad behoren aan het eind van het logisch ontwerp de transaktieschema's op te vragen. Samen met de resultaten van Transaktie analyse leveren ze een goed inzicht in de toekomstige situatie. Transaktieschema's kunnen vaak best door gebruikers worden opgesteld. In eerste instantie gaat het bij het maken ervan zoals bij elke vorm van ontwerpen: iets proberen, het weggooien, iets anders bedenken, op papier zetten, verbeteren, toch weer weggooien en opnieuw beginnen. Tenslotte ontstaat dan toch nog maar een voorlopig transaktieschema.

- Het maken van het startontwerp.

Meestal is dit het huiswerk voor de informatie-analist. In een wereld waar overal microcomputers staan, is het natuurlijk helemaal niet vreemd dat een gebruiker zelf zijn schermen ontwerpt. Een goede dialoogsimulator is voldoende gebruikersvriendelijk om dat mogelijk te maken.

- De evaluatie van het startontwerp.

De gebruiker beoordeelt de schermlay-outs, stelt wijzigingen voor en vraagt naar alternatieve oplossingen.

- De simulatie.

Dit is de belangrijkste stap in het hele gebeuren. De gebruiker heeft brondokumenten bij zich en gebruikt die zoals op het transaktieschema is aangegeven. De werkelijkheid moet zo nauwkeurig mogelijk worden gesimuleerd. Als het gaat om een loketsituatie moet iemand fungeren als klant, als het gaat om een transaktie waarin de telefoon een rol speelt, dan moet er worden opgebeld door een collega die weet hoe een gesprek verloopt. Als er nog cijfers moeten worden verzameld kan nu bijvoorbeeld gemeten worden hoe lang een transaktie duurt. Iedere transaktie wordt minstens 10 keer precies zo uitgevoerd als op het transaktieschema staat aangegeven. Het zou namelijk niet de eerste keer zijn dat een aardig dialoogontwerp na 10 keer toch niet zo handig blijkt te zijn. De gebruiker moet zich vooral niet laten verleiden slechts de getoonde schermen qua lay-out te beoordelen. De hele transaktie moet getest worden op gebruikersvriendelijkheid. Deze simulatie kan natuurlijk herhaald worden voor verscheidene groepen gebruikers. Bij grote aantallen mensen zullen de gebruikers zelf tot een selektie moeten komen.

- De evaluatie.

Aan het einde van alle simulatierondes kunnen de definitieve transaktieschema's worden gemaakt. De schermlay-outs zullen al tijdens de simulatie zijn aangepast en zo niet dan wordt nu de definitieve lay-out vastgesteld. De kans bestaat natuurlijk dat verschillende groepen gebruikers niet tot overeenstemming kunnen

komen. Het gebeurt vaak dat door dialoogsimulatie blijkt dat er in verschillende kantoren anders gewerkt wordt. Die verschillen kunnen funktioneel zijn, maar het kan ook om wildgroei gaan. Verschillende transakties voor hetzelfde doel betekenen extra ontwikkelingskosten, extra beheer, extra onderhoudskosten enzovoort. Op basis van het kostenplaatje moet de gebruikersorganisatie beslissen of er verschillende versies van de transakties moeten worden ontworpen. Uiteindelijk is ook een gebruiker gediend met een korte ontwikkelingstijd en een vlotte service tijdens het onderhoud. Vaak zal het neerkomen op kiezen uit de transaktie-ontwerpen.

- De overdracht van dokumenten.

De gebruikers ontvangen een kopie van elk transaktieschema en elke schermlay-out. Die zijn niet alleen bedoeld voor de aksepta-tietest maar ook om al gebruikt te worden voor het opstellen van gebruikershandleidingen, het doen van voorstellen voor organisatorische wijzigingen, het opleiden van gebruikers en het maken van nieuwe taak/funktie-omschrijvingen. Bij de opleiding van gebruikers kan natuurlijk heel goed gebruik gemaakt worden van de dialoogsimulator en de ontworpen transakties. Het is natuurlijk verstandig om hier pas mee te beginnen wanneer er geen grote wijzigingen meer worden verwacht.

Een andere term die in dit verband wel eens gebruikt wordt is prototyping. Men ontwerpt een prototype van het uiteindelijke systeem, dat wil zeggen dat men in principe het systeem al bouwt, maar alleen in grote lijnen. Er worden programma's en bestanden gemaakt, maar bijvoorbeeld niet alle kontroles en foutsituaties zijn er in opgenomen. Het is de mooi-weerkant van het uiteindelijke systeem. Er zijn echte programma's, die bijvoorbeeld berekeningen korrekt uitvoeren, er zijn echte bestanden aanwezig en helaas moet er ook een echte computer aanwezig zijn om het prototype op te bouwen. In situaties waarin al een computer aanwezig is van het type dat nodig is, is het geen probleem. Dan komen er alleen wat kleinere verschillen met dialoogsimulatie naar voren. Bij dialoogsimulatie zijn wijzigingen veel sneller en gemakkelijker door te voeren.

De dialoogsimulator is een onafhankelijke portable microcomputer die op elke werkplek in elke vestiging kan worden neergezet zonder dat er verder technische voorzieningen nodig zijn.

In nieuwbouw situaties, waar de computer nog moet worden aangeschaft, is de computer nodig om te kunnen prototypen. Sommige computerleveranciers hebben op hun computer hulpprogramma's beschikbaar om snel en gemakkelijk programma's en bestanden te kunnen ontwikkelen voor protoyping. Die hulpprogramma's dienen dan als verkoopargument en dat betekent dat toch de computer is gekocht voor dat de gebruiker zich een mening heeft kunnen vormen

over zijn toekomstige situatie. Een weg terug of overschakelen naar een ander fabrikaat voor het definitieve systeem is in de praktijk niet haalbaar. Bij de methode dialoogsimulatie wordt als middel een dialoogsimulator gebruikt. In de volgende paragraaf wordt het middel behandeld.
Zowel bij prototyping als bij dialoogsimulatie werkt de gebruiker met het beeldscherm. Bij prototyping komen de gegevens op het scherm uit echte bestanden via echte programma's. Voor de gebruiker is dat natuurlijk niet zichtbaar. Voor gebruikers is het verschil tussen beide methoden minimaal.

63.2 De dialoogsimulator

Zoals in Fig. 61.1 in beeld is gebracht komt het werken met beeldschermen neer op het voeren van een dialoog met de computer. Een transaktie is het geheel van handelingen inclusief het voeren van de dialoog. Per interaktie voert de computer opdrachten uit die resulteren in gegevens op het scherm. Dat kan een schermlayout zijn, maar ook een berekende verkoopprijs of een overzicht van de lopende orders. Hoe de computer die gegevens maakt is voor de dialoog niet interessant. De dialoog bestaat uit intypen, wachten en lezen wat er gedisplayd wordt.
Als op het transaktieschema is aangegeven dat de gebruiker een ordernummer wil intypen en dat de computer dan de orderregels op het scherm moet zetten, dan is daarmee de dialoog gedefinieerd. Onmisbaar onderdeel van de dialoog is vervolgens de schermlayout: waar moeten de regels op het scherm komen en wat moet de lay-out van iedere regel zijn? Waar de computer die regels vandaan haalt en wat er allemaal nog moet gebeuren voor hij ze op het scherm zet is voor de gebruiker die de regels wil bekijken niet van belang.
Een dialoogsimulator is een microcomputer die voor de gebruiker de funktie heeft van een beeldscherm zoals dat later op zijn bureau kan staan.
Aan de hand van de dialoog, zoals die is vastgelegd op het transaktieschema worden schermlay-outs gemaakt. Als die allemaal zijn gemaakt tijdens in de samenwerking tussen gebruikers en informatie-analisten kunnen ze direkt "live" geprobeerd worden. Er hoeven niet eerst programma's te worden ontwikkeld of bestanden te worden gevuld met gegevens. Als er op een bepaalde plaats iets ingetypt moet worden kan de gebruiker dat nu doen. Als de computer dan een orderregel moet displayen dan verschijnt er nu een orderregel volgens de vastgestelde lay-out. Laten we eens aannemen dat per regel moeten verschijnen: aantal, omschrijving, prijs per stuk. Alles wat de simulator moet displayen komt uit een standaardvoorraad gegevens. Daarin bevinden zich getallen, artikelom-

schrijvingen, bedragen, telefoonnummers enzovoort. Wanneer een orderregel moet worden gedisplayd verschijnt onder aantal een getal, onder artikel een artikelomschrijving en onder prijs per stuk een bedrag. Wat er op het scherm verschijnt is niet een korrekte prijs per stuk maar een willekeurig bedrag uit de gegevensvoorraad. De gebruiker ziet dus de dialoog volledig werken, alleen de inhoud van de velden zijn willekeurig. De gegevensvoorraad kan echter heel gemakkelijk worden aangepast zodat bijvoorbeeld artikelomschrijvingen een beetje passen bij het bedrijf.
Voor de gebruikers heeft de dialoogsimulator dus de funktie van een beeldscherm, in werkelijkheid is het een microcomputer. De dialoogsimulator hoeft dus niet precies gelijk te zijn aan het beeldscherm dat uiteindelijk op het bureau komt te staan. Funktioneel zijn alle beeldschermen hetzelfde. De lay-out van het toetsenbord verschilt nogal per fabrikant, maar bijna altijd zijn dezelfde standaardfunkties mogelijk. De verschillen betreffen hoofdzakelijk aspekten die voor de automatiseerders van belang zijn. In de praktijk blijkt steeds weer, dat gebruikers er geen probleem mee hebben dat een toets op een bepaalde plaats zit, maar dat ze niet weten wanneer ze hem moeten indrukken. Met andere woorden, dat het toetsenbord van een dialoogsimulator er wat anders uitziet dan dat van de uiteindelijke beeldschermen is geen echt probleem. Dialoogsimulatie wordt uitgevoerd tijdens het logisch ontwerp en de bouw van het systeem duurt meestal meer dan een half jaar. Welke gebruiker zou na een half jaar nog weten op welke toetsen van de dialoogsimulator hij heeft gedrukt? Nee, tijdens de akseptatietest maakt de gebruiker kennis met het definitieve toetsenbord en voert vervolgens de dialoog uit zoals die is vastgelegd op het transaktieschema.
Mocht er iets niet mogelijk zijn op de echte beeldschermen, dat wel is toegezegd tijdens de dialoogsimulatie, dan moeten de automatiseerders dat melden tijdens het technisch ontwerp. Daarom is het belangrijk dat gebruikers aan het eind van het technisch ontwerp ervoor zorgen dat automatiseerders niet geruisloos aan de bouw beginnen, maar eerst komen vertellen of alle transaktieschema's, schermlay-outs en toegezegde responsetijden nog gelden.
Behalve als gereedschap voor de methode dialoogsimulatie, is de simulator natuurlijk ook een uitstekend hulpmiddel in kursussen of bij begeleiding van gebruikers, die voor het eerst in aanraking komen met beeldschermen.

63.3 Responsetijden

Bij een rustige konversatie duurt het wachten op een antwoord enkele seconden. Als de gesprekspartner steeds binnen een halve seconde al zou antwoorden voelt men zich na een paar interakties

zeer ongelukkig. De partner luistert helemaal niet, of als zijn antwoorden het tegendeel bewijzen, is hij superintelligent. Een partner die konstant 10 seconden nodig heeft om een antwoord te bedenken, is ook zeer irritant. De wachttijden, die we in de automatisering responsetijden noemen, moeten ook binnen zekere grenzen liggen.
Stel dat een typiste moet werken met een typemachine of een tekstverwerker die aan het eind van iedere regel het toetsenbord blokkeert gedurende een tijd tussen 0 en 5 seconden. Soms kan ze gewoon doortypen, soms moet ze even wachten, soms 5 seconden. Er is dus geen sprake meer van een werkritme: met een dergelijk apparaat is niet te werken.
Als de wachttijd aan het einde van de regel nul is wanneer ze normaal typt, maar 5 seconden op het moment dat ze een liniaal moet oppakken, een blad moet omslaan en de liniaal weer neer moet leggen, is de situatie al beter. Er bestaat verband tussen responsetijden en de handelingen die moeten worden verricht. Het beeldscherm lijkt erg veel op een typemachine met wachttijden. Gedurende de responsetijd kan er meestal niet worden ingetypt, omdat we de reaktie van de computer nodig hebben om door te gaan. Hoe meer de computer moet doen, hoe langer de responsetijd wordt. Nu zijn computers enorm snel en ze kunnen de ingewikkeldste berekeningen in een fraktie van een seconde uitvoeren. Er zijn echter twee soorten problemen. Het eerste probleem is dat de meeste gegevens op magnetische schijven zijn opgeslagen en dat het de computer tijd kost die gegevens op te halen. Als een gebruiker 20 orderregels op z'n scherm wil zien en de computer zou die regels stuk voor stuk moeten ophalen dan kan dat wel 2 seconden duren. Het tweede probleem is dat de computer vele beeldschermen tegelijk moet bedienen. Hoe meer beeldschermen er zijn aangesloten, hoe moeilijker de computer het krijgt. Iedere computer kan maar een beperkt aantal beeldschermen aan. Hoe groot dat aantal is, is niet gemakkelijk te zeggen omdat het van de soort transakties afhangt die worden uitgevoerd. Transakties waarbij maar af en toe iets wordt ingetypt zijn ideaal voor een computer. Beeldschermen waar konstant blind wordt ingetypt met 4 aanslagen per seconde vormen een heel andere belasting.
De bepaling van responsetijden is het werk van specialisten. Het beheer van responsetijden begint bij de gebruikers die eisen stellen. Daarmee stuiten we meteen op een aantal problemen. Hoe kan een gebruiker eisen stellen, wat zijn redelijke eisen en wanneer moeten die eisen gesteld worden?
Soms worden eisen gesteld in algemene termen: de gemiddelde responsetijd moet minder zijn dan 2 seconden. Is dat het gemiddelde van alle mogelijke interakties van alle mogelijke transakties? Zo ja, dan is het vervelend als 20% van alle transakties in 80% van

de gevallen in gebruik is en dat in die transakties nu net de lange responsetijden zitten. Geen enkele automatiseerder kan iets doen met een dergelijke eis. Soms werken 10 ontwerpers aan een projekt, wie is er dan verantwoordelijk voor de gemiddelde responsetijd?
De eisen moeten dus specifiek en realistisch zijn. Dat wil zeggen dat, zeker van de meest gebruikte transakties, per interaktie een konkrete eis gesteld moet worden. Een realistische eis past bij het werkritme maar dat is niet op papier te zien, helemaal niet door mensen die nog nooit achter een beeldscherm hebben gezeten. Het werkritme van een nog te realiseren situatie kan alleen worden vastgesteld door de toekomstige situatie nauwkeurig te simuleren. Niet even met de dialoogsimulator kijken of de schermlayout akseptabel is, maar een echte werkomgeving scheppen en echt aan 't werk gaan met het beeldscherm. Dan zal blijken dat bij de ene interaktie een responsetijd van 1 seconde noodzakelijk is en bij een andere een van 10 seconden nog voldoet. Er bestaat verband tussen responsetijden en de handelingen die moeten worden verricht.
Op de dialoogsimulator kan voor iedere interaktie de responsetijd worden ingesteld. Die tijd moet door de informatie-analist worden ingesteld. Hij kan dus zonder moeite een overbelast systeem simuleren en de responsetijden laten toenemen tot de gebruiker ze niet meer akseptabel vindt. Ervaren informatie-analisten zijn in staat om na het opstellen van het transaktieschema al aan te geven welke responsetijden waarschijnlijk problemen zullen geven. Zij kunnen dat duidelijk maken door op de simulator tijden te kiezen van bijvoorbeeld 5 seconden. De gebruiker krijgt dan een beeld van de manier van werken als het systeem inderdaad zo gaat werken.
Responsetijden tijdens het logisch ontwerp voorspellen kan niemand, maar uitzonderlijke situaties moeten herkend worden. Bij minder ervaren informatie-analisten of nieuwe, onbekende computersystemen is er nog niets aan de hand als gebruikers dan maar aan het eind van het technisch ontwerp met de ontwerpers alsnog de transakties doornemen om vast te stellen of de eisen gehaald zullen worden.
Zo niet, dan heeft de gebruiker de keus uit 3 mogelijkheden:
- de langere responsetijden aksepteren
- de transaktie laten vervallen
- een andere transaktie ontwerpen.

Ontwerpen is immers een iteratief proces?
Nogmaals, niemand kan voorspellen of een responsetijd 1 of 1 1/2 seconde wordt. In de praktijk leveren zulke verschillen leveren geen probleem op. Aan het eind van het technisch ontwerp moet bekend zijn of het om 1 of om 4 seconden gaat.

Als achteraf blijkt dat responsetijden te lang zijn dan wordt dat veroorzaakt door ontwerpfouten of door een te kleine, overbelaste computer. Pogingen om die situaties te voorkomen beginnen bij de gebruikers: konkrete eisen stellen aan responsetijden, realistische cijfers verstrekken aan de informatie-analisten en het eind van het technisch ontwerp in de gaten houden. Als door onjuiste gegevens van de gebruikers later het aantal beeldschermen moet worden uitgebreid moet men zich er niet over verbazen dat de computer dat niet meer aan kan en dat de responsetijden oplopen. Kortom, het beheer van responsetijden begint bij specifieke, realistische eisen van de gebruikers en bij korrekt cijfermateriaal.

63.4 Eisen gevraagd

In het verleden hadden gebruikers het relatief gemakkelijk. De informatie-analisten analyseerden de bedrijfsprocessen en de gegevens die daarbij gebruikt werden. Een jaar later bleek een computer een deel van de gegevens op te nemen en lijsten voor de gebruikers te produceren. Je kon dan als gebruiker kritiek leveren op de informatie die op die manier door het computersysteem werd verstrekt. Die aanpak van de automatisering was mogelijk, omdat het bijna altijd ging om administratieve processen, die de informatie-analisten soms beter beheersten dan de gebruikers zelf. Bovendien ging het niet om interaktieve toepassingen en hield het werken met lijsten de afstand tussen gebruiker en computer groot. Tegenwoordig gaat het bijna altijd om interaktieve toepassingen en om veel meer zaken dan de administratieve standaardprocessen. Dat betekent dat de inbreng van de informatie-analist veel minder zal worden en dat de gebruiker moet zeggen wat hij nu eigenlijk wil.

We moeten daarbij een aantal zaken gescheiden houden. In de eerste plaats de drie aspekten van het werken met computers: de gegevens waar het systeem mee werkt, de berekeningen die moeten worden uitgevoerd en de dialoog die wordt gebruikt. In de tweede plaats het verschil in gebruikers. Het zal vaak gebeuren dat een gebruiker gegevens invoert, maar geen idee heeft van de verwerking van die gegevens. Dat bepaalde gegevens terechtkomen in een bestand, dat door andere funktionarissen via een beeldscherm bekeken kan worden, interesseert hem absoluut niet. Een afdelingschef die via een beeldscherm aan het eind van de dag wil zien wat de voorraad in guldens is, zal precies moeten aangeven hoe de berekening daarvan moet worden uitgevoerd.

Vaak blijkt dat gebruikers die alle processen uitstekend beheersen, het moeilijk krijgen als ze moeten aangeven wat ze nu eigenlijk precies op het scherm willen zien. Degelijk transaktie-ontwerp begint bij het transaktieschema. Daar wordt de aansluiting

op de andere handmatige procedures duidelijk. De menselijke handelingen van de gebruiker tijdens de transaktie worden vastgelegd. Het transaktieschema is eigenlijk het dokument waarop de gebruiker in zijn taal, eisen stelt aan het te ontwerpen systeem. Daarom moeten gebruikers het opstellen van transaktieschema's eisen van de informatie-analisten. De dialoogsimulator levert natuurlijk niets op ten aanzien van de berekeningen die moeten worden uitgevoerd. Toch blijkt vaak dat een gebruiker, die achter een beeldscherm wordt geplaatst, heel snel tot uitspraken komt over wat hij nu eigenlijk wil. Met een dialoogsimulator kan hij gemakkelijk iets proberen, iets laten zien aan collega's, een alternatief proberen of iets nieuws bedenken. En daarmee zijn we bij de kreatieve gebruiker aangekomen.

63.5 De kreatieve gebruiker

In de meeste systeemontwikkelingsmethoden is geen plaats voor een kreatieve gebruiker. Informatie-analisten brengen volgens een of andere methode de bedrijfsprocessen en gegevens in kaart, om een deel van de processen geautomatiseerd te laten verlopen. In principe wordt altijd de bestaande situatie geanalyseerd en geautomatiseerd. Gebruikers die nog nooit achter een beeldscherm hebben gezeten kunnen niet met kreatieve plannen komen. Die komen pas als het systeem is gebouwd en dan is het voorlopig te laat. De kreativiteit van gebruikers kan alleen worden ingeschakeld als ze tijdens het logisch ontwerp zien hoe een beeldscherm funktioneert op hun werkplek. Als ze zien hoe gemakkelijk het is op een simulator alternatieven te proberen ontstaat ook de sfeer om met een nieuw idee te komen en het meteen aan collega's te demonstreren. Het is gemakkelijk om automatisering de schuld te geven van veel werkeloosheid. Automatisering doet banen verdwijnen maar ook verschijnen en verschuiven. Dat verschuiven kan nieuw werk betekenen, bedacht door kreatieve gebruikers. Een taak verandert van inhoud doordat routinewerk door de computer wordt gedaan en er tijd vrij komt voor nieuwe werkzaamheden, die het bedrijf ten goede komen. Een regel is daar niet voor te geven, wel zijn nu methoden besproken die de omgeving scheppen om gebruikers kreatief te laten omgaan met veranderingen die de computer met zich meebrengt.
Het heeft wat dit betreft geen zin naar de automatiseerders te kijken. Die hebben hun handen vol om de automatisering op zich goed te laten funktioneren.
Gebruikers moeten alleen toepassingen van methoden eisen die ook funktioneel kreatieve geesten de ruimte geeft. Daarbij blijft natuurlijk gelden dat aan alles een prijskaartje hangt. Nieuwe aktiviteiten scheppen kost iets en moet dus direkt of indirekt een

bijdrage leveren tot de doelstellingen van het bedrijf.

Hoofdstuk 64
Transaktie analyse

64.1 Transaktie analyse als methode

Transaktie analyse is een methode om eisen van gebruikers in cijfers uit te drukken en om te rekenen naar konsekwenties voor gebruikers en het computersysteem. Ook hier is weer sprake van een methode. Volgens vastgelegde regels wordt het transaktieschema dat gebruiker en informatie-analist hebben gemaakt, door een transaktie-analist vertaald in cijfers. Op die manier ontstaat een transaktieschema in cijfers. Als erop staat dat er een naam moet worden ingetypt, dan noteert de transaktie-analist het aantal toetsen dat moet worden aangeslagen. Als erop staat dat de gebruiker een nummer op het scherm moet vergelijken met een nummer op een bon, dan bepaalt de transaktie-analist de tijd die daarvoor nodig is. Het rekenprogramma dat deze gegevens verwerkt levert cijfers op die voor gebruikers van belang zijn en cijfers die de automatiseerders gebruiken bij het technische ontwerp van het computersysteem.
Er bestaan drie vormen Transaktie analyse: de ergonomische, de logische en de technische analyse. In Fig. 64.1 is de samenwerking tussen gebruikers en automatiseerders in beeld gebracht. Tijdens het logisch ontwerp kan een ergonomische en/of een logische Transaktie analyse worden uitgevoerd. Voor de gebruikers is de inbreng hetzelfde. Wanneer tijdens het technisch ontwerp een

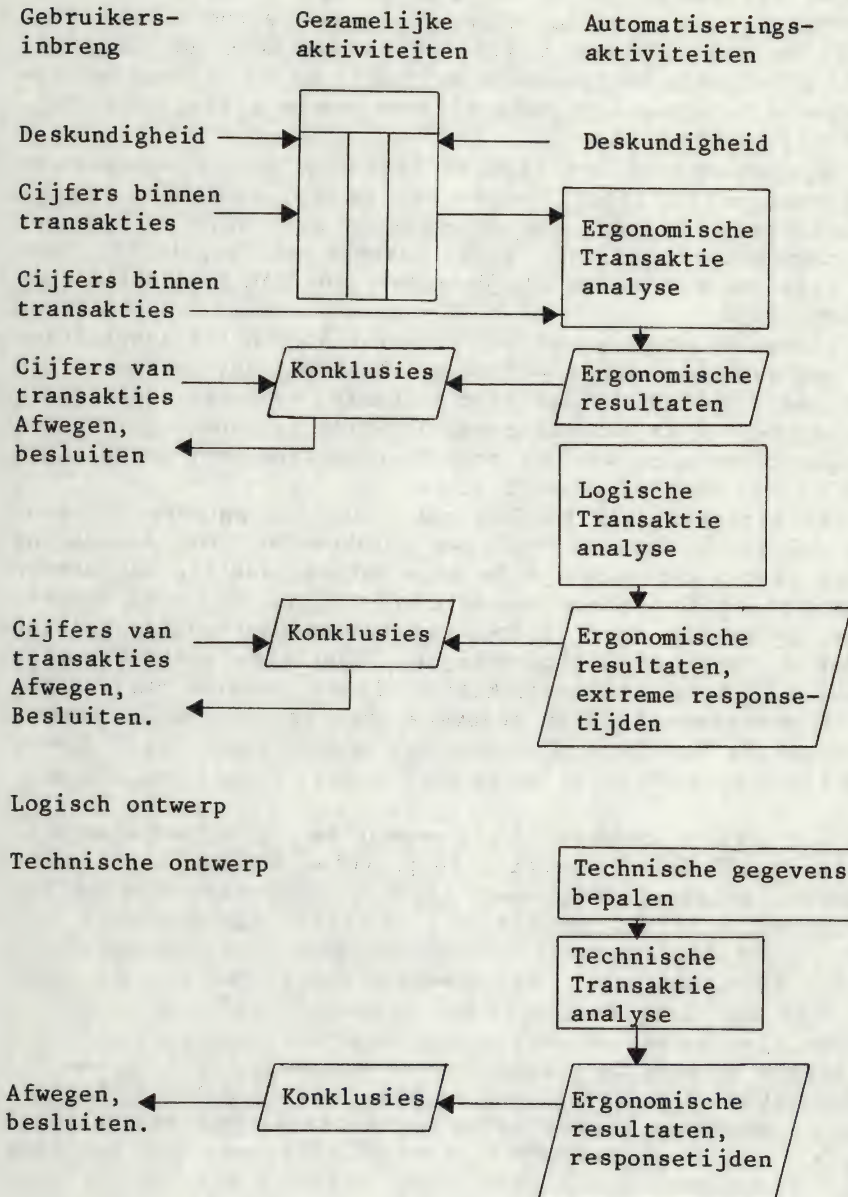


Fig. 64.1 De samenwerking rond Transaktie analyse.

technische Transaktie analyse wordt uitgevoerd dan is dat alleen een uitbreiding van de beide andere analyses. De toevoeging bestaat uit de detaillering van de verwerking door de computer. Daar zijn geen gebruikersgegevens bij nodig en de tijdens het logisch ontwerp verstrekte cijfers blijven gewoon geldig en worden gebruikt bij de berekening.
Cijfers binnen transakties zijn cijfers die op het transaktieschema kunnen worden gezet: lengte van velden, aantallen tekens, aantallen regels. Cijfers van transakties zijn cijfers over het aantal transakties per dag, pieksituaties en dergelijke. Deze cijfers zijn pas van belang als de resultaten van Transaktie analyse bekend zijn.
Cijfers binnen transakties worden verwerkt door het rekenprogramma, cijfers van transakties worden gebruikt bij het berekenen van de konklusies. Die konklusies zijn de basis voor het besluitvormingsproces binnen de gebruikersorganisatie: ze kunnen leiden tot voorlopige goedkeuring van het ontworpen systeem, tot aanpassing van de eisen of van het ontwerp.
Dat proces herhaalt zich tijdens het technisch ontwerp: de konklusies leiden dan tot definitieve goedkeuring, tot aanpassing van eisen of van het ontwerp. De ergonomische analyse kan worden uitgevoerd zodra de dialoogsimulatie voorbij is. Bij deze analyse wordt de verwerking door de computer buiten beschouwing gelaten en worden er eenvoudig responsetijden ingesteld, zoals die tijdens de dialoogsimulatie noodzakelijk bleken. Na deze analyse beschikt de gebruiker al over gegevens over aantal beeldschermen, aantal uren per beeldscherm enzovoort. Deze analyse sluit direkt aan bij dialoogsimulatie en wordt uitgevoerd tijdens het logisch ontwerp.
Tijdens het logisch ontwerp is een moment aan te wijzen waarop de verwerking van iedere interaktie in principe bekend is. De automatiseerders zeggen dan dat het logisch gegevensontwerp of het gegevensmodel gereed is. Nu kan de transaktie-analist ook de verwerking in principe door het rekenprogramma laten omrekenen in verwachte responsetijden. De nauwkeurigheid daarvan is niet groot, maar het gaat erom uitzonderlijke situaties nu reeds te signaleren. Dit noemen we de logische Transaktie analyse. Tijdens het technisch ontwerp is precies bekend hoe de verwerking van iedere interaktie wordt uitgevoerd en kan de transaktie-analist het programma nauwkeurig genoeg laten berekenen wat de responsetijd zal worden. Bij deze berekeningen wordt uitgegaan van een niet overbelaste computer. Als gebruikers bijvoorbeeld opeens meer beeldschermen wensen dan oorspronkelijk was gepland kan de computer daardoor overbelast raken en kunnen de responsetijden hoog oplopen.
Tijdens het technisch ontwerp kunnen responsetijden redelijk

Terminal Transaktie tijd (T.T.T.)

Rubriek	Tussen-waarden		Aantal seconden	Variantie
Invoerlengte	76.03			
Intiksnelheid	2.50			
	--------	/		
Invoertijd			30.41	192.46
Wachttijden en denktijden			3.00	3.00
Responses	7.71			
Transportvertraging	1.20			
	--------	*		
Transporttijd			9.25	61.22
Response-eenheden	24.01			
Tijd per eenheid	0.10			
	--------	*		
Verwerkingstijd			2.40	5.85
Uitvoerlengte	208.74			
Afdruksnelheid	25000.00			
	--------	/		
Uitvoertijd			0.01	0.00
			----------	----------
Subtotaal			45.08	262.53
Foutherstelpercentage	5.00			
Foutherstel-tijd			2.25	1.14
Aanloop en uitloop			2.50	0.60
			----------	----------
Netto T.T.T.			49.83	264.27
(Volledig effektief)	Standaardafwijking			16.26
Bruto T.T.T.			71.18	642.74
(Bij 70% effektief)	Standaardafwijking			25.35

Procentuele verdeling van de T.T.T.

Invoertijd	61.03 %
Wachttijden en denktijden	6.02 %
Dialoogresponse-tijden	18.98 %
Afsluitresponse-tijden	4.42 %
Uitvoer-tijd	0.02 %
Tijd voor fout herstellen	4.76 %
Tijd voor aanloop en uitloop	5.02 %

Fig. 64.2 De pagina Terminaltransaktietijd.

nauwkeurig berekend worden. Gebruikers moeten er dus voor zorgen dat aan het eind van het technisch ontwerp gekontroleerd wordt of de tijdens de dialoogsimulatie gespecificeerde eisen gehaald worden. Zo niet dan moet de oorzaak daarvan achterhaald worden. Misschien hebben de gebruikers eisen hebben gesteld die met het geplande computersysteem niet goed zijn te realiseren. Dat kan betekenen dat er een andere transaktie moet worden ontworpen op de bekende manier. Ontwerpen is toch een iteratief proces?
De basis voor Transaktie analyse is het transaktieschema. Als het goed is, is de definitieve vorm ervan ontstaan door dialoogsimulatie. Maar ook al zou de dialoogsimulatie om de een of andere reden niet zijn uitgevoerd, dan is Transaktie analyse nog altijd mogelijk. Met andere woorden, gebruikers kunnen te allen tijde de beschikking krijgen over de resultaten van Transaktie analyse tijdens het ontwerpen.

64.2 Resultaten van Transaktie analyse

Transaktie analyse rekent het transaktieschema om naar resultaten in cijfers. Het transaktieschema wordt tijdens het logisch ontwerp gemaakt. De resultaten komen voor de gebruikers dus ook tijdens die fase van de projektaanpak beschikbaar. Het rekenprogramma levert drie pagina's met resultaten per transaktie. Iedere transaktie is daarmee kwantitatief gekarakteriseerd. De tweede pagina is weergegeven in Fig. 64.2.
Op deze pagina wordt de terminaltransaktietijd (T.T.T.) berekend. Er wordt een netto en een bruto T.T.T. bepaald. Stel dat de tijd die nodig is om een transaktie uit te voeren 60 seconden bedraagt. Een simpele redenering zou dan zijn, dat er maar een beeldscherm nodig is om per achturige werkdag 480 transakties te verwerken: 480 x 60/(8 x 3600) = 1. In de praktijk gaat het niet zo, omdat niemand precies 8 uur per dag werkt. Er bestaan dingen als pauzes, persoonlijke verzorging en sociale kontakten. Deze tijd moet worden omgeslagen over de uit te voeren transakties, en wel door een effektiviteitspercentage in te voeren. Dit is het percentage van de uren per dag dat er effektief gewerkt wordt achter het beeldscherm. Een gemiddeld percentage is 70%. Als de netto tijd om een transaktie uit te voeren 60 seconden duurt, bedraagt de bruto terminaltransaktietijd 85,7 seconden. Dan zijn er dus 480 x 85,7/(8 x 3600) = 1.43 beeldschermen nodig. In de praktijk worden dat er dus 2.
Men zou dus kunnen bezuinigen op het aantal beeldschermen door de effektiviteitspercentage op 80 of 90 te stellen. De gebruikers werken dan dus anders dan in het gemiddelde bedrijf, wat heel goed mogelijk is. In sommige bedrijven wordt in ploegen gewerkt: naast een groep beeldschermen is een zitje voor de rustende

ploeg. In zo'n geval wordt het effektiviteitspercentage 100, omdat er achter het beeldscherm geen koffie wordt gedronken maar intensief wordt gewerkt.
Het effektiviteitspercentage wordt door de gebruikers vastgesteld. De informatie-analist noteert slechts. Zoals blijkt uit de pagina met resultaten, is altijd direkt te zien met welk percentage het programma heeft gerekend. In het voorbeeld is dat 70%. Op de pagina T.T.T. van het rekenprogramma is verder precies te zien hoe de berekende terminaltransaktietijd is samengesteld. De samenstellende elementen zijn deels gebaseerd op gebruikersgegevens, deels op technische gegevens van automatiseerders. Bij een ergonomische analyse zijn die gegevens niet verwerkt, maar is de transaktie-analist uitgegaan van bepaalde responsetijden. Bij een logische analyse zijn de technische gegevens nog onnauwkeurig, bij een technische analyse zijn ze gebaseerd op de machine waarop het systeem wordt gebouwd. We zullen nu de samenstellende elementen van de T.T.T. doornemen.
- Invoertijd. Dit is de tijd die nodig is om alle gegevens in te typen. Daarin is verwerkt het "soms" en "af en toe" van het transaktieschema. De tijd wordt berekend door het totaal aantal aanslagen te delen door de typesnelheid. Het aantal aanslagen hangt af van de dialoog zoals die samen met de informatie-analist is ontworpen en is vastgelegd op het transaktieschema. De typesnelheid is een gegeven dat wordt verstrekt door de gebruikers. Iemand met een typediploma moet 2 aanslagen per seconde halen. In situaties waarin gebruikers niet gewend zijn te typen, kan aan de informatie-analist gevraagd worden om de transakties ook nog door te rekenen met een lagere typesnelheid. Dan wordt de T.T.T. langer en dat betekent in de aanloopfase meer beeldschermen of overwerken of werk verschuiven naar de volgende dag.
- Denk- en wachttijden. Dit is het totaal van de tijd die nodig is om te lezen, te kontroleren, te vergelijken. Deze tijden zijn tijdens het maken van het transaktieschema in overleg bepaald of gemeten tijdens dialoogsimulatie en hangen af van de gekozen procedure en de ontworpen dialoog.
- Transporttijd. De responsetijd bestaat soms uit twee delen. Als de beeldschermen via een telefoonlijn of een netwerk verbonden zijn met de elders opgestelde computer is het ene deel de tijd die nodig is om een bericht naat de compter te sturen en om een bericht terug te krijgen, het andere deel is altijd de verwerkingstijd door de computer. De transporttijd is de totale tijd die nodig is om alle berichten van de transaktie van en naar de computer te transporteren. Deze tijd hangt af van de ontworpen dialoog, maar ook van een aantal technische zaken. Als er geen netwerk of telefoonlijn in geding is, is deze tijd nul.
- Verwerkingstijd. De responsetijd bestaat geheel of gedeeltelijk

uit de tijd die nodig is om het bericht te verwerken. Dat bericht kan zijn het ingetypte klantnummer, waarbij de computer naam, adres en woonplaats moet zoeken en terug moet sturen. De verwerkingstijd is de som van de verwerkingstijden van alle interakties. De som van de transporttijd en de verwerkingstijd is de tijd die de gebruiker totaal wacht op antwoord van de computer oftewel de som van alle responsetijden. In het algemeen zal dat maar een klein deel van de T.T.T. uitmaken. Deze tijden zeggen niets over de responsetijden. Als de som 20 seconden bedraagt kan het gaan om 2 responsetijden van 10 seconden, maar even goed om 10 van 2 seconden.

- Uitvoertijd. Dit is de tijd die nodig is om gegevens af te drukken met een printer. Printers kunnen op allerlei manieren worden ingezet. Het voert te ver om dat nu te behandelen. Voor de gebruiker is het soms van belang de printtijd te vergelijken met de T.T.T. Als een groep van 5 beeldschermen is uitgerust met een gemeenschappelijke printer, dan is het van belang dat de automatiseerders voorrekenen hoeveel tijd het printen kost, in hoeverre de printer in de pas blijft lopen met de beeldschermen en hoe de responsetijden zullen worden tijdens het printen.
- Fouthersteltijd. Dit is de tijd die nodig is om fouten te herstellen. Het gaat niet om aanslagfouten, maar om fouten die gemaakt worden als gevolg van de lengte van rubrieken. Een kodenummer van twaalf cijfers intypen vergt koncentratie. Het foutherstelpercentage hangt af van de rubrieklengte en de typesnelheid. Het percentage wordt vastgesteld door de transaktie-analist.
- Aan- en uitloop. Dit is de tijd die voorafgaat aan de dialoog met de computer resp. de tijd die erop volgt. Bij een lokettransaktie op een station is het aanhoren van de bestemming aanloop en het overhandigen van het kaartje en het afrekenen, uitloop. Bij transakties waarin een telefoon wordt gebruikt, is ook heel duidelijk sprake van aan- en uitloop. Het is duidelijk dat de aan- en uitloop belangrijk zijn voor de computer: hoe meer aan- en uitloop hoe minder hij per beeldscherm heeft te doen en hoe meer beeldschermen hij aankan.

De aan- en uitloop hangen af van de gekozen procedure, vaak met name van de aansluiting op andere handmatige procedures. Ook deze aspekten moeten in cijfers worden uitgedrukt. Soms is schatten erg moeilijk, maar korrekte simulatie van de transaktie met een dialoogsimulator maakt meting met een stopwatch mogelijk.

Als nu op basis van de T.T.T. het aantal beeldschermen wordt berekend en dat aantal veel te groot blijkt, dan kan het zijn dat we een andere transaktie of een andere dialoog moeten ontwerpen. Ontwerpen is toch een iteratief proces? De cijfers zijn tijdens het logisch ontwerp bekend, dus moet er nog gelegenheid zijn een paar dingen opnieuw te doen. De samenstellende elementen van de

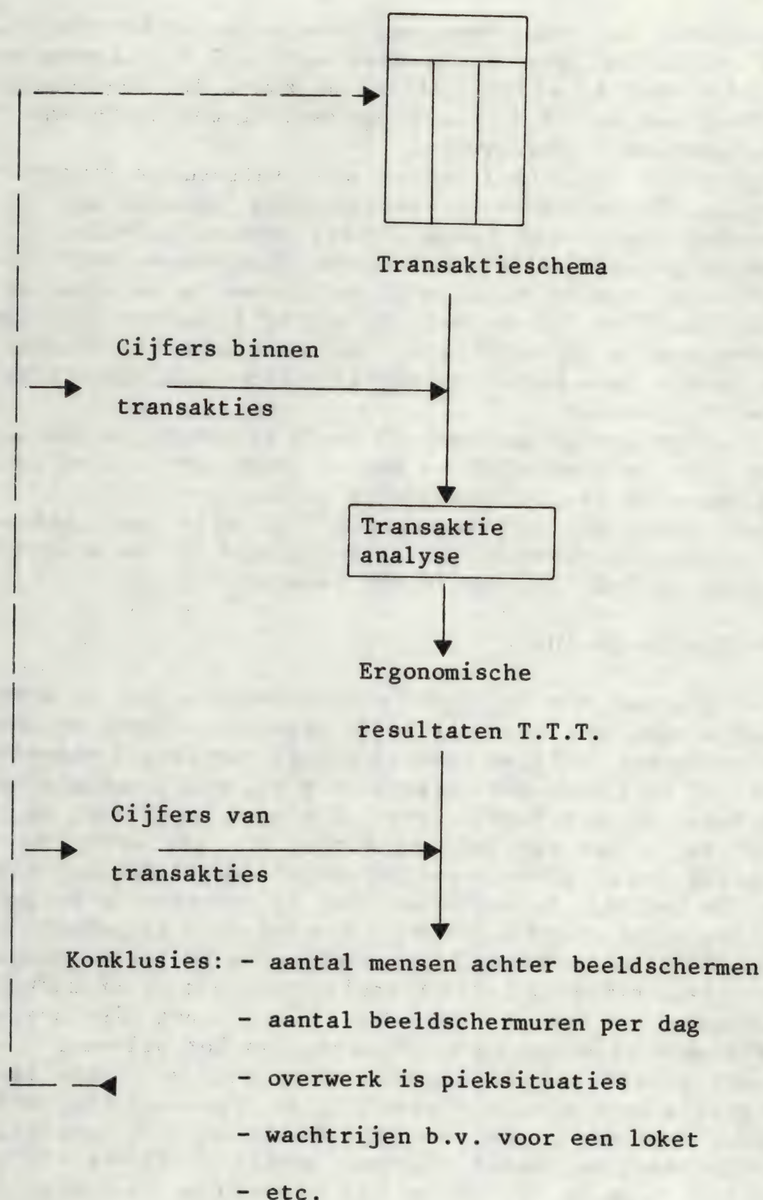


Fig. 64.3 Van cijfers naar cijfers.

T.T.T. geven direkt aan waar eventueel iets verdiend kan worden. Als de denk- en wachttijden 1% uitmaken van de T.T.T. hoeven we daar verder niet naar te kijken. Alleen de zaken die een belangrijk deel vormen van de T.T.T. kunnen in overleg met de informatie-analisten opnieuw bekeken worden.
Zoals te zien is in Fig. 64.2 worden de resultaten weer uitgedrukt in een gemiddelde waarde en een spreiding. Daarmee kan bijvoorbeeld worden uitgerekend hoeveel beeldschermen er nodig zijn om in 95% van de gevallen alle transakties af te kunnen handelen. De gebruiker dient zich te realiseren dat termen op het transaktieschema als "soms" en "af en toe", op zichzelf reeds aanleiding geven tot spreiding in de resultaten. De resultaten van Transaktie analyse in deze vorm maken het mogelijk een aantal praktijksituaties door te rekenen.
In Fig. 64.3 is het geheel nog eens in beeld gebracht. De gestippelde pijlen geven de mogelijkheden aan om iteraties te starten, wanneer de konklusies daartoe aanleiding geven.
In het deel voor de informatie-analist zijn een aantal voorbeelden gegeven van gebruik van de resultaten, in de volgende paragraaf zullen we daar nog enkele aan toevoegen.

64.3 Sociale aspekten in cijfers

Er wordt veel gepraat over de sociale konsekwenties van de automatisering. Met name over werk dat wordt overgenomen door de computer, waardoor banen verloren gaan of geheel van inhoud veranderen. Niemand kan iets doen met de konklusie van dit soort algemene beschouwingen. Of taak/funktie-omschrijvingen veranderen en zo ja, in welke mate, kan pas bekeken worden als per werkplek is geinventariseerd welke procedures vervangen kunnen worden door transakties. De individuele werknemer moet de gelegenheid krijgen die transakties te ervaren, voordat er een computer is gekocht of een informatiesysteem voor hem is gebouwd op de bestaande computer. Daarvoor is dialoogsimulatie bedoeld. Maar als de transaktie op zich akseptabel is voor de gebruiker wil dat nog niet zeggen dat alle problemen zijn opgelost. Misschien is het helemaal niet akseptabel dat iemand van 's morgens vroeg tot 's avonds laat achter het beeldscherm zit, hoe prachtig de transakties op zich ook zijn. Daar komt een aantal sociale aspekten om de hoek kijken. Transaktie analyse levert daarvoor konkrete cijfers. In de vorige paragraaf zijn de resultaten per transaktie besproken. In de praktijk zullen er per werkplek vaak verschillende transakties worden uitgevoerd. Per werkplek is bekend welke transakties er worden uitgevoerd. De informatie-analist maakt aan de hand van de T.T.T.'s van alle transakties een overzicht als weergegeven in Fig. 64.4. Dit overzicht geeft aan om welke werkplekken het gaat,

Tijdbestedingdiagram

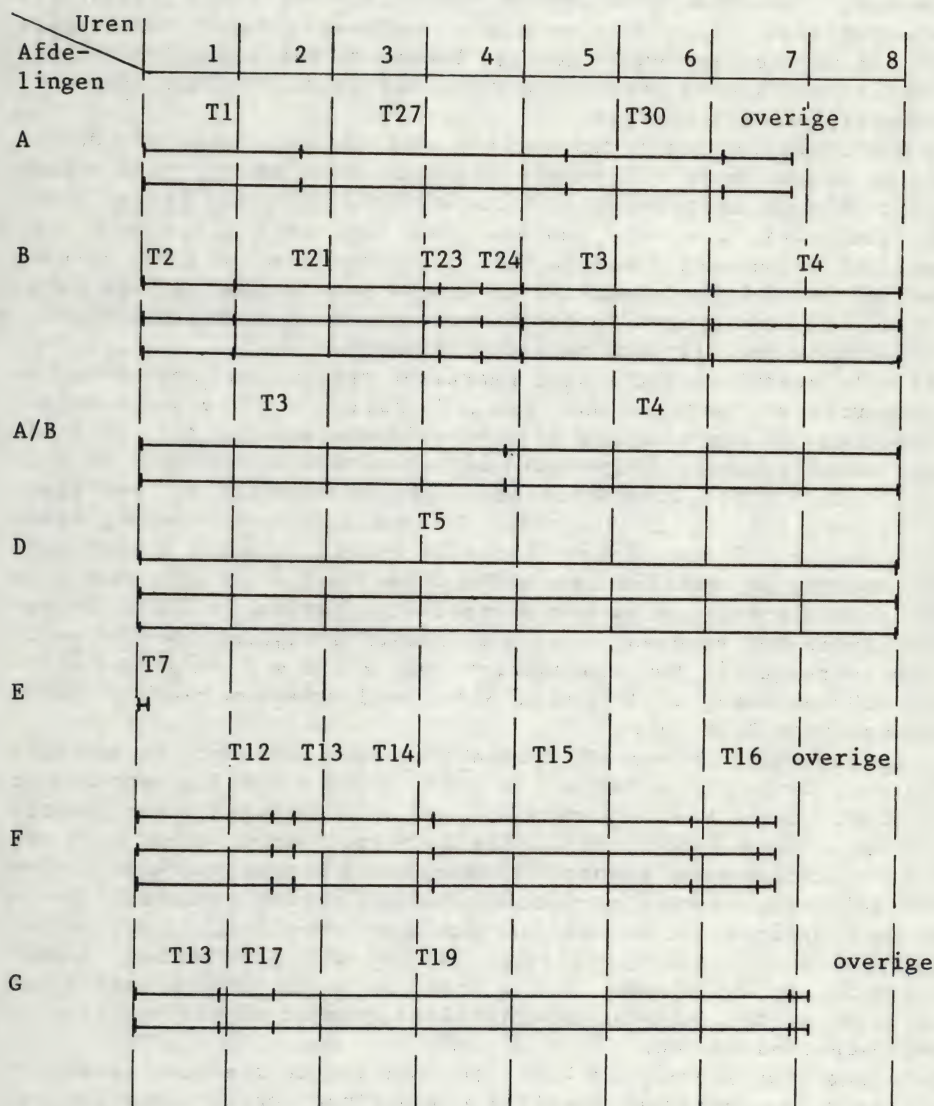


Fig. 64.4 Het tijdbestedingsoverzicht.

welke transakties op iedere werkplek worden uitgevoerd en hoe lang dat duurt.
Daarbij valt het volgende te merken. In dit voorbeeld is aangenomen dat per afdeling het aantal transakties gelijkmatig over de aanwezige beeldschermen worden verdeeld. Dat hoeft natuurlijk helemaal niet. Toen de gebruikers aangaven hoeveel transakties per dag moesten worden uitgevoerd hadden ze die aantallen ook per werkplek aan kunnen geven. Dan wordt het aantal beeldschermuren per werkplek verschillend.
Op dit overzicht komen transakties voor die vele uren per dag in beslag nemen, maar ook enkele die maar heel weinig tijd nemen. Dat zijn vaak transakties die qua cijfers niet van belang zijn. Een transaktie die maar een keer per dag wordt uitgevoerd, mag best een responsetijd hebben van 10, 20 seconden. Bij het uitvoeren van een enkele transaktie is immers geen sprake van een werkritme. In het algemeen is het onevenredig duur om dit soort uitzonderingen ook flitsend te laten verlopen.
Uit zo'n overzicht kan iedere gebruiker opmaken om hoeveel beeldschermuren per werkplek het gaat. We zullen een voorbeeld verder uitwerken. Op een afdeling zijn 4 beeldschermen gepland. Er wordt maar een transaktie uitgevoerd: het intoetsen van orders. De bruto T.T.T. van die transaktie bedraagt 120 seconden met een standaardafwijking van 20 seconden. De werktijd wordt zodanig berekend dat in 95% van de gevallen alle orders verwerkt moeten kunnen worden. De werktijd is $n \times T.T.T. + 2 \times \sqrt{n \times var}$ waarin n het aantal orders is en var de variantie is van de T.T.T. De variantie is het kwadraat van de standaard afwijking. Dus de werktijd bedraagt bij 900 transakties: $900 \times 120 + 2 \times \sqrt{900 \times 400} =$ 109.200 seconden, of 30,3 uur. Bij 4 beeldschermen betekent dat per werkplek 7,58 uur.
Op de piekdagen worden 20% transakties meer verwacht. De werktijd bedraagt dan: $1080 \times 120 + 2 \times \sqrt{1080 \times 400} =$ 130.914 seconden of 36,4 uur. Bij 4 beeldschermen moet er op zo'n dag 9,1 uur gewerkt worden. Anders gezegd: bij 4 beeldschermen komen we op zo'n dag 4,4 beeldschermuren tekort. De gebruikers kunnen ook kiezen voor een oplossing waarbij er 5 beeldschermen worden geplaatst. Op de normale dagen wordt dat beeldscherm niet gebruikt, op de piekdagen 4,4 uur. Gemiddeld blijven er bij 4 beeldschermen, zonder overwerk, op de piekdag 4,4 x 3600/120 = 132 orders liggen tot de volgende dag. Als dat akseptabel is, kunnen we het op die 4 beeldschermen houden.
Door het kwantitatief in kaart brengen van de transakties worden dit soort berekeningen mogelijk tijdens het logisch ontwerp. Dat is meestal ook vroeg genoeg om allerlei organisatorische wijzigingen voor te bereiden.

64.4 Vergissingen van gebruikers

In deze paragraaf gaat het om redeneringen of denkbeelden van gebruikers, die achteraf in de praktijk, onherroepelijke vergissingen bleken te zijn. Er is in de automatisering geen weg terug en daarom is het effekt van die vergissingen definitief. Het is voor alle gebruikers die nog aan het begin staan, belangrijk zich af te vragen of sommige van hun meningen ook tot die vergissingen kunnen behoren.

- Automatiseren is een zeer technisch vak, waar je als leek toch nooit verstand van krijgt en hoe zou je je daar dan mee moeten bemoeien.

Automatisering is inderdaad een vak apart, maar de gebruiker moet zich alleen bemoeien met het gebruik van de automatisering. Hij moet aangeven hoe hij met het beeldscherm wil werken. De eenvoudigste typiste kan toch ook beoordelen of een typemachine of tekstverwerker prettig te bedienen is of niet?

- Automatisering is een zaak van automatiseerders. De direktie heeft zeer deskundige mensen aangesteld of ingehuurd en die gaan een systeem ontwikkelen, dat precies bij het bedrijf past en waardoor het werk vlotter en beter gaat.

Hoe deskundig automatiseerders ook zijn, gebruikers kennen hun eigen manier van werken het beste. Daarom moeten gebruikers er voor zorgen dat er volgens kontroleerbare, overeengekomen methoden overlegd wordt over het te ontwerpen systeem. Het ontwerp van het systeem kan noch gedelegeerd worden aan interne noch aan externe automatiseerders.

- Automatisering kun je tegenhouden door je er als gebruiker buiten te houden door zo weinig mogelijk gegevens te verstrekken, door geen tijd te hebben etc.

Of automatisering in een bepaalde situatie tegen te houden is, is de vraag, maar in ieder geval niet doordat gebruikers zich er niet mee bemoeien. In dat geval ontwerpen de automatiseerders het systeem alleen, naar hun eigen beste weten en met eigen schattingen en gegevens. Dat systeem wordt gebouwd en ingevoerd, ook bij de meest onwillege gebruiker.

- Er is geen tijd beschikbaar om aan automatisering te besteden.

Er zijn inderdaad talloze gebruikers met overvolle agenda's en lange werkdagen. Maar eigenlijk komt gebrek aan tijd neer op geen tijd hebben voor de eigen nieuwe werksituatie of die van de medewerkers. Nu zijn inderdaad niet alle automatiseerders even vlot in het afhandelen van gesprekken met gebruikers, zodat alles erg veel tijd kost. Wat het ontwerpen van transakties betreft kan men met de besproken methoden in ieder geval snel en effektief werken.

- Automatisering is een rijdende trein die niet meer te stoppen

is.
Het stoppen of bijsturen van de automatisering blijkt in sommige gevallen mogelijk maar dan alleen als er tijdens het logisch ontwerp keiharde feiten op tafel liggen bijvoorbeeld in de vorm van cijfers. Dat is iets anders dan vage, softe diskussies over veranderingen, plezier in het werk, kontakten met collega's en dergelijke. Niet, dat die diskussies geen bijdrage kunnen leveren in de uiteindelijke vorm van de automatisering, maar cijfers die aantonen dat het werken met beeldschermen meer tijd kost dan de huidige procedures, zijn veel sprekender. Cijfers in rapporten zijn zeer effektief. Transaktie analyse levert die cijfers.
Het is zinloos te praten over de gevolgen van automatisering in het algemeen en triest te moeten praten over de slecht verlopen geschiedenis van de automatisering in de eigen omgeving. Die geschiedenis moet door gebruikers zelf gemaakt worden door aktief en kreatief te zijn tijdens het ontwerp.

Synoptische inhoud per paragraaf

Deel 1, voor het strategisch management.

11.1 Enige problemen.
Hoewel er bedrijven zullen zijn waar het, volgens het management, prima gaat met de automatisering, zullen velen een of meer van de geschetste problemen herkennen.

11.2 Waarom methoden?
Veel van de geschetste problemen vinden hun oorzaak in de gebrekkige kommunikatie tussen gebruikers en automatiseerders. De oplossing moet dus gezocht worden in het werken volgens gekozen methoden.

11.3 Maatwerk of konfektie?
Het ontwerpen van interaktieve toepassingen is maatwerk. De mate waarin het strategisch management zich ermee bemoeit past eerder bij de aanschaf van een kant en klaar produkt.

11.4 De ivoren toren van Babel.
De toren van Babel is het symbool van de spraakverwarring, de ivoren toren dat van de zelfgenoegzaamheid van ontwerpers. De ivoren toren van Babel is de automatisering in veel bedrijven: onbegrijpelijk, ongrijpbaar en kostbaar.

Er zijn methoden nodig om tijdens het logisch ontwerp gebruikers en automatiseerders rond het beeldscherm te krijgen. Dialoogsimulatie zorgt voor de kwalitatieve aspekten van het ontwerp, Transaktie analyse voor de cijfers, zowel wat de gebruikers als wat de technische aspekten betreft.

11.5 De zweep erover?
Gebruikers en automatiseerders kiezen gezamenlijk voor methoden, vanwege de resultaten en de aansluiting op de denk- en werkwijze van beiden. Daarna is er geen diskussie meer mogelijk over het wel of niet toepassen van die gekozen methoden.

11.6 Sociale aspekten van de automatisering.
Verhalen over chips en werkeloosheid lossen de problemen van de individuele werknemer niet op. Een kursus burgerinformatica lost de problemen met de professionele automatisering in bedrijven niet op. Gebruikers opleiden tot mede-ontwerpers maakt hen tot mondige gebruikers, die ook begrip krijgen voor de problemen die ze zelf veroorzaken.

12.1 Decentralisatie.
Afgezien van technische of organisatorische argumenten blijkt decentralisatie soms te worden ingegeven door de zucht van gebruikers naar een stuk zelfstandigheid. Ze willen uit onder de macht van de grote automatiseringsafdeling. In die gevallen kan het middel erger blijken dan de kwaal.

12.2 Microcomputers.
Wat gebeurt is bij minicomputers, gebeurt weer bij de microcomputers, maar nu op een veel grotere schaal. Men zoekt onafhankelijkheid van de automatiseringsspecialisten, wordt gemotiveerd tot de aanschaf van een microcomputer door verkopers en komt na korte tijd tot de ontdekking dat het middel erger is dan de kwaal.

12.3 Kantoorautomatisering.
De algemeen bekende gegevensverwerking zal eens een geintegreerd onderdeel uitmaken van kantoorautomatisering. Waar kantoorautomatisering verder ook uit zal bestaan, er zal altijd gezocht moeten worden naar middelen die passen bij de behoeften. Methoden om de gegevensverwerking te beheersen zullen daarom altijd nodig blijven.

12.4 Beveiliging en uitwijk.
De meeste systemen zijn slecht beveiligd, evenals vele bedrijven. Als de dreiging toeneemt zal de behoefte aan beveiliging toenemen

zelfs als daar ongemak voor legale gebruikers tegenover staat. Het gaat om de afweging van kosten en ongemak tegen de kans op fraude en de gevolgen. Bepaalde kalamiteiten kunnen afdoend worden opgevangen door computeruitwijk.

12.5 Strategische vergissingen.
In veel bedrijven verdiept het strategisch management zich te weinig in een bedrijfsonderdeel dat steeds meer gaat kosten: automatisering. Daarnaast blijkt men de eigen organisatie vaak slecht te kennen. Dat leidt tot een verkeerde aanpak en tot te globale beslissingen.

12.6 Strategische moed.
Op strategisch niveau heeft automatisering niets meer met techniek te maken en is daar dus relatief eenvoudig. Er moeten beslissingen genomen worden die de beschreven problematiek in verband met automatiseringsprojekten, voor een belangrijk deel zouden oplossen.

Deel 2, voor het taktisch gebruikersmanagement.

21.2 Vier x M.
Het vakmanschap van de gebruiker wordt gekoppeld aan het vakmanschap van de informatie-analist via methoden. Per methode kunnen middelen, tools of gereedschappen beschikbaar zijn. Last but not least komt het management van het geheel aan de orde.

21.2 Wie zijn de gebruikers, wie de informatie-analisten?
Er wordt een viertal types gebruikers gedefinieerd en twee soorten informatie-analisten. Voor de verschillende gebruikers zal de samenwerking met de informatie-analisten ook verschillen.

21.3 Kommuniceren met automatiseerders.
Waarover spreken zij in de verschillende fasen? Alsof automatisering nog niet moeilijk genoeg is, blijken sommige cijfers en uitspraken bedrijfspolitiek gezien, moeilijk te liggen.

21.4 Waarom methoden?
De enige manier om vat te krijgen op de automatisering is methoden kiezen die de kommunikatie tussen gebruikers en informatie-analisten regelen. Toepassing van methoden betekent planbare standaardaktiviteiten en kontrole aan de hand van standaarddokumenten.

21.5 Eigen betrokkenheid.
In hoeverre moet een taktisch gebruikersmanager betrokken zijn

bij de automatisering? Niet door een hoeveelheid technische kennis op te doen, maar door goed om te gaan met de fasen van een projektaanpak en methoden voor de kommunikatie met automatiseerders. Dat betekent soms het opleggen van een stuk discipline aan gebruikers.

22.1 Fasen.
De fasen van een projektaanpak worden behandeld. Per fase wordt aangegeven wat er van de gebruikers verwacht wordt en welke initiatieven ze moeten ontplooien.

22.2 De scharnierpunten.
In aansluiting op de vorige paragraaf wordt nu aangegeven hoe belangrijk voor de gebruikers het logisch ontwerp wordt, als de juiste methoden worden toegepast en wat er aan het eind van het technisch ontwerp moet gebeuren voordat met de bouw wordt begonnen.

22.3 Interaktieve toepassingen.
Het verschil tussen het ontwerpen van batch toepassingen en van interaktieve toepassingen wordt beschreven. Veel systeemontwikkelingsmethoden zijn nog te veel gericht op de batch-toepassingen en er worden geen methoden aangegeven voor de realisering van de inbreng van gebruikers.

22.4 Dialoogsimulatie.
De methode wordt kort beschreven met de te bereiken resultaten. Hier valt ook het woord responsetijden. De dialoogsimulator wordt alleen genoemd als gereedschap.

22.5 Transaktie analyse.
Met Transaktie analyse kunnen tijdens het logisch ontwerp de uiteindelijke situaties van de gebruikers al worden doorgerekend naar aantallen beeldschermen, beeldschermuren, overwerk, pieksituaties enzovoort. Kortom, sociale aspekten in cijfers.

22.6 Responsetijden.
Wat zijn responsetijden, hoe ontstaan ze en waardoor worden ze beinvloed? Getracht wordt voor gebruikers duidelijk te maken dat het beheer van responsetijden door de automatiseerders begint met konkrete cijfers, uitspraken en eisen van gebruikers.

23.1 Sociale aspekten.
In algemene termen wordt er door iedereen gepraat over de sociale aspekten van de automatisering. Voor een bepaalde gebruiker worden de sociale aspekten pas konkreet als hij hoort dat hij met

een beeldscherm moet gaan werken. Hoewel uren per dag? Hoe moeilijk is het? Wat blijft er van mijn funktie over? Met toepassing van de juiste methoden weet de gebruiker dat voordat de computer is gekocht.

23.2 Microcomputers en kantoorautomatisering.
Bij interaktieve toepasingen denkt men meestal aan beeldschermen en computers. De methoden voor het ontwerpen van interaktieve toepassingen slaan dan ook meestal op gegevensverwerking. Kantoorautomatiseringsfunkties worden meestal kant en klaar gekocht. Netwerken zullen echter altijd moeten worden ontworpen. Voor gebruikers is dat niet van belang, ze blijven echter verantwoordelijk voor het cijfermateriaal.

23.3 Taktische problemen.
De problemen die het taktisch gebruikersmanagement heeft met het managen van de automatisering betreffen: het stellen van eisen, het inzicht in het eindresultaat, de kommunikatie met automatiseerders, het beheer van de voortgang en het werken met standaard, pakketten.

23.4 Taktische fouten.
In deze paragraaf wordt een opsomming gegeven van in de praktijk veel voorkomende fouten. Deze fouten frustreren automatiseerders, bemoeilijken de voortgang van een projekt en zijn dus uiteindedelijk nadelig voor de gebruikers zelf.

23.5 Adviezen.
Ten aanzien van de in de vorige paragraaf genoemde fouten wordt een aantal adviezen gegeven ter voorkoming ervan. Een deel van die adviezen is alleen bruikbaar wanneer de behandelde methoden voor het ontwerp worden toegepast.

Deel 3, voor het taktisch automatiseringsmanagement

31.1 Vier x M.
In deze paragraaf gaat het over het vakmanschap van de mensen, het toepassen van methoden, het gebruik van gereedschappen en de invloed van het management op het geheel.

31.2 Methoden en omgevingen.
Automatisering wordt ingevoerd in grote en in kleine bedrijven, met grote mainframes en met minicomputers. Bij projekten gaat het om nieuwbouw, om uitbreidingen of konversies. In deze paragraaf wordt in grote lijnen het gebruik van de te behandelen methoden aangegeven voor de verschillende omgevingen.

31.3 Systeemontwikkelingsmethoden en de witte vlekken.
Bestudering van systeemontwikkelingsmethoden en de evaluatie van vele projekten tonen aan dat in de algemeen bekende systeemontwikkelingsmethoden twee witte vlekken aanwezig zijn: een waar het gaat om de kommunikatie met de gebruikers en een waar het gaat om het ontwerpen van de konfiguratie en het netwerk.

31.4 Vijf soorten vakmanschap.
Bij het beheer van responsetijden blijken vijf soorten specialisten nodig te zijn en een beheerder die ze met elkaar verbindt. In de praktijk blijkt vaak dat lange responsetijden gemakkelijk hadden kunnen worden voorkomen als de kommunikatie tussen de specialisten tijdens het ontwerp beter was geweest.

31.5 Aanpak.
Hoewel het om methoden gaat die van wezenlijk belang zijn voor gebruikers, zal toch vaak de aanzet van de kant van de automatisering moeten komen. Gebruikers staan zover af van de automatisering dat het stellen van eisen aan hen verkocht zal moeten worden.

32.1 Wie zijn de gebruikers?
De indeling van de gebruikers in drie groepen heeft alles te maken met de toe te passen methoden. Per type gebruiker wordt aangegeven wat er moet gebeuren om een keer van de steeds terugkerende problemen in de automatisering af te komen.

32.2 Problemen laten waar ze horen.
Automatiseerders zijn niet aangesteld om sociale of bedrijfspolitieke problemen op te lossen. Die moeten gesignaleerd worden. Wanneer de methoden goed worden toegepast kan er misschien toch nog het een en ander worden opgelost. Desnoods kan het projekt gerealiseerd worden ondanks weerspannige gebruikers.

33.1 Commercie en techniek.
Hoewel het om de bepaling van een konfiguratie gaat blijkt dat in de praktijk nu eenmaal niet de grootste rol te spelen. In de meeste gevallen kan dat ook niet omdat bedrijven hun problemen niet kunnen stellen in de termen die computerleveranciers zeggen nodig te hebben voor de bepaling van een konfiguratie.

33.2 Performance van interaktieve toepassingen.
Wiskundige benadering van de performance van computersystemen wordt al gauw onoverzichtelijk ingewikkeld. Toch beschikken computerleveranciers over meetgegevens waarvan enkele parameters dezelfde zijn als sommige resultaten van Transaktie analyse. Een

evaluatie van dit lijkt gewenst. Voor vergelijkende metingen van de belasting van transakties op een gegeven systeem is in ieder geval voldoende aanleiding.

33.3 Het volgende performance-debâcle.
Veel managers hopen dat de techniek nog eens zover zal komen dat we door goedkoper wordende hardware nog eens zoveel kapaciteit ter beschikking krijgen, dat ze in een keer van alle zorgen af zijn. De praktijk bewijst het tegenovergestelde: eerst met de minicomputer, nu met de microcomputer.

34.1 Transaktie als entiteitstype.
Interaktieve systemen worden gebruikt voor het uitvoeren van transakties aan beeldschermen. De handelingen van de gebruiker zijn van belang voor het ontwerp, maar ook voor de uiteindelijke systeembelasting. Daarom is ontwerpen van programma's die dialogen afhandelen onvoldoende. In verband met systeembeasting, konfiguratiekeuze en netwerkontwerp is het van belang transakties als entiteitstype in te voeren.

34.2 Waarom transaktie-ontwerp?
Transaktie-ontwerp maakt het mogelijk gebruikers in te schakelen als mede-ontwerpers en zo ook een stuk van de verantwoordelijkheid te laten dragen. Zowel kwalitatief als kwantitatief krijgt de gebruiker tijdens het logisch ontwerp inzicht in de uiteindelijke situatie, en legt de automatiseerder de basis voor een goed technisch ontwerp.

34.3 Transaktie-ontwerp en Transaktie analyse.
Binnen het transaktie-ontwerp zorgt de methode Transaktie analyse voor de berekening van de kwantitatieve aspekten van het ontwerp.

34.4 Transaktie-ontwerp en andere ontwerpaktiviteiten.
Transaktie-ontwerp vervangt geen enkele aktiviteit die gericht is op gegevens- en programma-ontwerp. De resultaten van transaktie-ontwerp moeten wel worden afgestemd op de resultaten van die aktiviteiten.

34.5 Transaktie-ontwerp en projektaanpak.
Per fase van een algemene projektaanpak wordt in grote lijnen de werkwijze besproken.

34.6 Transaktie-ontwerp en systeemontwikkelingsmethoden.
In veel ontwikkelingsmethoden wordt aangegeven dat er gekommuniceerd moet worden met gebruikers, maar die kommunikatie is gericht op de automatiseerders: het verkrijgen van goedkeuring voor

schermlay-outs en dialoogontwerp. Kwantitatieve aspekten van het ontwerp ontbreken altijd.

34.7 Transaktie-ontwerp in het grote geheel.
Problemen rond de sociale gevolgen van de automatisering, de veranderingen voor de organisatie, de gegevensdistributie en het netwerkontwerp kunnen met elkaar in verband gebracht worden op een konkrete, te beheren wijze door transaktie-ontwerp.

34.8 De benodigde tijd.
Hoewel er geen gemiddelde projekten, transakties of schermlay-outs bestaan wordt toch een poging gedaan wat richtgetallen te geven voor de benodigde tijd voor dialoogsimulatie en Transaktie analyse. In de praktijk zal die tijd per projekt moeten worden geschat door informatie-analisten en transaktie-analisten.

34.9 Responsetijden en transaktie-ontwerp
Voor de meest irritante, meest voorkomende en duurste fout in de automatisering bestaan geen konkrete eisen. In deze paragraaf gaat het over het verwerkingsdeel van de responsetijd. Bij te zwak projektmanagement kan men voor verrassingen komen te staan, waarvan iedereen zich achteraf afvraagt waarom ze niet eerder gesignaleerd zijn. Voor responsetijden ligt het scharnierpunt aan het eind van het technisch ontwerp. Responsetijden in de analysefase, tijdens het logisch ontwerp en tijdens het technisch ontwerp.

35.1 Waarom dialoogsimulatie?
Eigenlijk zouden gebruikers aan automatiseerders uit moeten kunnen leggen waarom dialoogsimulatie moet worden toegepast als onderdeel van transaktie-ontwerp. Toch heeft de methode ook een aantal konkrete voordelen voor de automatiseerders. Beide partijen zouden gemotiveerd moeten zijn om via dialoogsimulatie samen te werken.

35.2 Dialoogsimulatie versus prototyping.
Voor gebruikers is het verschil tussen dialoogsimulatie en prototyping erg klein, voor automatiseerders zijn beide methoden niet zo goed vergelijkbaar. Dialoogsimulatie heeft als voordeel dat het altijd en veel gemakkelijker inzetbaar is. Dialoogsimulatie kan daarom ook dienen als voorportaal voor prototyping.

36.1 Waarom Transaktie analyse?
De resultaten van Transaktie analyse vormen de argumenten om de analyse uit te voeren.

36.2 Transaktie analyse in grote lijnen.
In het kort wordt aangegeven uit welke aktiviteiten Transaktie analyse bestaat en welke funktionarissen er bij betrokken zijn.

36.3 Kwantiteiten.
In de automatisering zijn we gewend aan logisch denken: 0/1, JA/ NEE, IF...THEN...ELSE. Gerekend wordt er niet vaak. Gebruikers zijn trouwens ook niet zo scheutig met cijfers. In deze paragraaf wordt aangegeven wat er met de cijfers gebeurt.

37.1 Wat zijn netwerken.
Voor een uitgebreide behandeling van allerlei soorten netwerken wordt verwezen naar de literatuur. De meest voorkomende vormen en hun specifieke problemen worden behandeld in het kader van de bepaling van de hoeveelheid verkeer.

37.2 Hoe worden netwerken ontworpen?
De theorie is meestal wel bekend, de praktijk is anders. Als de systeemontwerpers niet beschikken over de gegevens die nodig zijn om de kapaciteit van een netwerk te bepalen, geeft de leverancier van de apparatuur een advies(!)

37.3 Netwerkontwerp in de vakliteratuur.
In vele handboeken worden de kwalitatieve aspekten van netwerken uitgebreid behandeld, de kwantitatieve aspekten worden behandeld op een manier die niet aansluit bij het ontwerp van transakties of de rest van het systeem.

37.4 Netwerkontwerp en Transaktie analyse.
Netwerkontwerpers hebben cijfers nodig die gebruikers noch systeemontwerpers kunnen verstrekken. In deze paragraaf wordt aangegeven hoe Transaktie analyse, als onderdeel van transaktieontwerp, als brug fungeert tussen gebruikersgegevens en netwerkontwerp.

37.5 Datanet 1 en Transaktie analyse.
In veel bedrijven wordt bij de beschouwingen over Datanet 1 teveel alleen op de direkte maandelijkse lasten gelet. Het totale kostenplaatje met apparatuur, personeel en de maandelijkse lasten dient de basis te zijn voor de afweging van Datanet 1 tegen een eigen netwerk. Daarnaast dienen ook nog moeilijk te kapitaliseren zaken als flexibiliteit bij verhuizingen en uitwijk te worden meegenomen in de beslissing. Transaktie analyse levert gegevens voor de variabele kosten per maand.

37.6 De invoering van micro's.
Microcomputers zullen maar op beperkte schaal stand alone blijven toegepast. Op de duur krijgen de meeste gebruikers behoefte aan gegevens uit het mainframe. Die kommunikatie moet weer worden ontworpen voor de meeste gegevensverwerkende toepassingen. LAN's leveren geen bijdrage in de funktionele koppeling.

37.7 Van informatiebehoeften naar netwerkontwerp.
In de inleiding zijn twee witte vlekken aangeduid in de systeemontwikkeling. In deze paragraaf wordt de weg van informatie-analyse ten dienste van het informatieplan naar het netwerkontwerp geschetst en gerelateerd aan de sociale, organisatorische en funktionele gevolgen van de automatisering.

37.8 Netwerkontwerp in distributieve omgevingen.
Het ontwerpen van distributieve systemen is een komplex geheel. De gebruikerseisen ten aanzien van de opslag van gegevens zijn vaak vaag. Meestal zijn er nogal wat alternatieven die tegen elkaar moeten worden afgewogen. In de literatuur wordt altijd het netwerk genoemd, maar er wordt nooit aangegeven hoe het ontworpen moet worden. In deze paragraaf wordt in grote lijnen aangegeven hoe in een distributieve omgeving transaktie-ontwerp leidt tot een netwerkontwerp. In de andere delen wordt de aanpak verder uitgewerkt.

Deel 4, voor informatie-analisten.

41.1 Taakomschrijving en vakmanschap.
De taakomschrijving van een informatie-analist is meestal erg vaag. Er zijn pogingen gedaan om tot standaardisatie te komen. Hoe het zij, in deze paragraaf worden de eigenschappen en kwaliteiten omschreven die nodig zijn om in ieder geval de beschreven methoden goed te kunnen uitvoeren.

41.2 Methoden en middelen.
De methoden en middelen die we in dit deel behandelen, worden hier in grote lijnen weergegeven. De iteratieve aspekten van het ontwerpproces worden terloops alvast aangestipt.

41.3 Relaties tussen methoden.
De te beschrijven methoden zijn nauw met elkaar verweven en strekken zich uit over logisch en technisch ontwerp. Wil iteratie mogelijk zijn dan moet de koppeling tussen de methoden worden aangegeven. Dat gebeurt in deze paragraaf.

41.4 Projektaanpak.
In de konkrete situatie waarin de informatie-analist aan het werk gaat bestaan al andere methoden. Soms ongestruktureerd en soms in het geheel van een systeemontwikkelingsmethode. De te beschrijven methoden moeten aansluiten op bestaande methoden. In deze paragraaf wordt die koppeling beschreven in algemene termen, maar per dokument van de te behandelen methoden.

41.5 Methoden en de omgeving.
Er is een duidelijk verschil tussen een automatiseringsafdeling van een groot concern en die van het midden- en kleinbedrijf. In een omgeving van data dictionaries, modellenbouw en specialisten op elk gebied ligt het ontwerpen anders dan wanneer een informatie-analist en een systeemontwerper samen het hele ontwerp op een minicomputer moeten trekken. In beide gevallen zijn de te beschrijven methoden uitstekend bruikbaar, maar de relatie met andere methoden is per definitie anders. In deze paragraaf wordt een indeling van omgevingen gemaakt.

42.1 Transaktie als entiteitstype.
In de meeste bedrijven zijn de interaktieve toepassingen een ongekwantificeerde kluwen van programma's en bestanden. Bij het in kaart brengen van gegevens worden entiteitstypes vastgesteld en attributen bepaald. Het geheel wordt nauwkeurig beschreven in recordlay-outs. Die nauwkeurigheid is noodzakelijk om de programma's te kunnen laten werken. Wanneer men ooit greep wil krijgen op zaken als performance, responsetijden, terminalbezetting en vele andere zaken, moet een transaktie als entiteitstype worden bepaald en moet een aantal attributen worden vastgelegd. In deze paragraaf worden ze beschreven en voor een deel besproken.

42.2 Van analyse naar transaktie-ontwerp.
Een belangrijke vraag bij de overgang van de analyse van de huidige situatie naar het ontwerp van de nieuwe situatie is: wanneer ontstaan transakties? Inkopen is meestal geen transaktie, maar het noteren van een prijs evenmin. Hoe gedetailleerd moet de analyse worden uitgevoerd om de vertaling naar transakties mogelijk te maken?

42.3 Transaktieschema's
Een transaktieschema beschrijft volgordelijk de procedure aan het beeldscherm. Vaak is de dialoog met de computer allesbehalve volgordelijk: er kan gesprongen worden naar andere schermen, sommige schermen worden herhaald. In deze paragraaf wordt aangegeven hoe transaktieschema's ook in komplexe situaties, gemaakt kunnen worden.

42.4 Transaktie-ontwerp in verschillende omgevingen
De inbreng van gebruikers is in kleine bedrijven met een minicomputer even belangrijk is in grote bedrijven met mainframes, een grote automatiseringsstaf en vele methoden en tools. De methoden om die inbreng te realiseren moeten in alle omgevingen aansluiten bij de manier van werken. In deze paragraaf wordt in diverse omgevingen die aansluiting aangegeven.

42.5 Geografie in analyse en ontwerp.
In een distributieve omgeving zijn gegevens vaak op een andere plaats opgeslagen, dan waar ze gebruikt worden. Een netwerk dient om de geografische verschillen tussen gebruik en opslag op te lossen. Wanneer bij transaktie-ontwerp ook de lokaties in kaart worden gebracht, ligt het gebruik in ieder geval vast. Een belangrijk aspekt bij het netwerkontwerp.

42.6 Transaktie-ontwerp en beeldschermontwerp.
Het enige dat automatiseerders boeit aan een beeldscherm is de dialoog. Schermlay-out en velddefinities zijn hun eerste en meestal ook hun enige interesse. Transaktie ontwerp begint anders.

42.7 Transaktie-ontwerp in distributieve omgevingen.
In omgevingen van distributieve processing en distributieve databases gaat het om komplexe zaken. We gaan voorbij aan alle bedrijfspolitieke problemen en stellen vast welke bijdragen vanuit het logisch ontwerp geleverd kunnen worden om in het technisch ontwerp tot een netwerkontwerp te komen.

43.1 Dialoogsimulatie als methode.
In achtien stappen wordt beschreven hoe de methode uitgevoerd moet worden om een goed resultaat te bereiken. Verschillen in omgeving spelen daarbij geen rol. De verschillen die de automatisering betreffen, zijn in het vorige hoofdstuk behandeld.

43.2 Problemen.
In deze paragraaf wordt een aantal problemen genoemd die kunnen voorkomen in het leven van de informatie-analist die in aanraking komt met dialoogsimulatie. De oplossing voor die problemen zal de informatie-analist bij zichzelf moeten zoeken.

43.3 De dialoogsimulator.
In deze paragraaf worden de belangrijkste aspekten van een dialoogsimulator besproken. Wil een simulator kunnen funktioneren binnen het transaktie-ontwerp zoals dat in dit boek wordt beschreven dan zullen deze funkties in ieder geval aanwezig moeten

zijn. In de praktijk blijkt wel dat het handig is om over nog wat nevenfunkties te beschikken.

43.4 Responsetijden.
In het kader van dit hoofdstuk gaat het om responsetijden tijdens de simulatie. Het beheer van responsetijden begint bij het stellen van konkrete eisen. In deze paragraaf wordt aangegeven welke bijdrage de simulator daarin levert.

43.5 Dialoogsimulatie als hulpmiddel voor de analysefase.
Hoewel dialoogsimulatie een ontwerpmethode is kan het gereedschap soms nuttig gebruikt worden tijdens de analysefase, om de gebruiker zich bewust te laten worden van bepaalde processen of procedures.

44.1 Transaktie analyse.
Deze paragraaf begint met een definitie van Transaktie analyse. Vervolgens worden de sleutelwoorden uit die definitie stuk voor stuk besproken.

44.2 Vormen van Transaktie analyse.
Hoewel er maar een rekenprogramma is, bestaan er toch drie vormen van Transaktie analyse: de ergonomische, de logische en de technische Transaktie analyse. De vorm wordt bepaald door de parameters die worden gebruikt. Welke vorm wordt gekozen hangt af van het doel dat men wil bereiken. In het algemeen heeft de informatie-analist alleen belangstelling voor de ergonomische vorm.

44.3 Kwantiteiten binnen transakties.
Het detailschema is het gekwantificeerde transaktieschema. Het transaktieschema beschrijft de procedure aan het beeldscherm. Om een detailschema te kunnen maken moeten er dus cijfers worden vastgesteld binnen de transaktie.

44.4 Parameters.
Op het detailschema wordt de procedure aan het beeldscherm in parameters beschreven. In deze paragraaf worden de parameters behandeld die van belang zijn voor de ergonomische Transaktie analyse. Per parameter wordt aangegeven hoe er gekwantificeerd moet worden.

44.5 Kwantiteiten van transakties.
Behalve kwantiteiten binnen transakties (44.3) bestaan er ook cijfers van transakties. Bij deze cijfers gaat het om bijvoorbeeld het aantal transakties per dag. Ze zijn nodig om de resultaten van Transaktie analyse te vertalen in konklusies.

44.6 Ergonomische resultaten en konklusies.
Omdat de ergonomische resultaten direkt betrekking hebben op de gebruiker, zal de informatie-analist deze resultaten vertalen in konklusies. In deze paragraaf wordt behandeld wat die resultaten precies zijn en welke konklusies eruit getrokken kunnen worden.

44.7 Voorbeelden van ergonomische konklusies.
In de vorige paragraaf zijn de ergonomische resultaten van Transaktie analyse behandeld, in deze paragraaf worden een paar praktijkvoorbeelden gegeven. In de voorbeelden komen de sociale aspekten in cijfers aan de orde.

Deel 5, voor transaktie-analisten.

51.1 Taakomschrijving en vakmanschap.
Een transaktie-analist is iemand die Transaktie analyse beheerst in alle opzichten. Als het gaat om de ergonomische resultaten zou een informatie-analist heel goed de transaktie-analist kunnen zijn. Als het gaat om de technische resultaten gaan de gedachten eerder uit naar een systeemontwerper. In het geval van netwerkontwerp op basis van Transaktie analyse is uiteraard kennis van datakommunikatie onontbeerlijk.

51.2 Transaktie analyse als methode.
Transaktie analyse in grote lijnen, de kommunikatie tussen informatie-analisten en transaktie-analisten en de vorm van Transaktie analyse in verschillende omgevingen.

52.1 Transaktieschema's.
Transaktieschema's vormen het interface-dokument tussen informatie-analist en transaktie-analist. In deze paragraaf wordt de ontvangst van de transaktieschema's door de transaktie-analist besproken.

52.2 Detailschema's.
Detailschema's zijn gekwantificeerde transaktieschema's. De indeling van het detailschema wordt behandeld. Tevens wordt aangegeven hoe de leesbaarheid verbeterd kan worden.

52.3 Resultaten van het rekenprogramma.
In het deel voor de informatie-analist zijn de eerste twee pagina's output van het rekenprogramma behandeld, in deze paragraaf wordt de derde en laatste pagina besproken. Hierbij gaat het om cijfers over verkeer en responsetijden.

52.4 Vormen van Transaktie analyse.
De drie vormen ergonomische, logische en technische Transaktie analyse worden behandeld. Van de ergonomische en de logische is een globale vorm mogelijk.

52.5 Omgevingen en soorten toepassingen.
Om niet steeds in de tekst de omgeving te hoeven omschrijven wordt hier een aantal typische omgevingen gedefinieerd: enerzijds de systemen en een onderlinge koppelingen, anderzijds de soorten projekten waarin Transaktie analyse kan worden uitgevoerd.

52.6 Verkeersparameters.
Per interaktie kan op het detailschema het verkeer worden aangegeven. Het rekenprogramma bepaalt dan de berichtlengte heen en terug en de tijd tussen de berichten. Dat zijn de basisgetallen voor alle verkeersberekeningen voor alle soorten netwerken. Zo kan het transportdeel van de responsetijden worden bepaald.

52.7 Verwerkingsparameters.
Responsetijden bestaan voor een deel uit verwerkingstijd. Op het detailschema kan die verwerking worden aangegeven per interaktie. Het programma berekent het verwerkingsdeel van de responsetijden.

52.8 Printparameters.
Hoewel het bij transakties meestal gaat om beeldschermtransakties, bestaan er natuurlijk ook nog keyboardprinters, printers gekoppeld aan beeldschermen, printers ten behoeve van een cluster van beeldschermen en printers per werkplek. In deze paragraaf geven we aan hoe in de genoemde situaties de printparameters kunnen worden gebruikt.

52.9 Benodigde tijd.
Hoewel het door de grote variatie in soorten transakties moeilijk is algemene cijfers te geven, worden hier toch wat richtgetallen gegeven voor het maken van de detailschema's.

53.1 Resultaten en konklusies.
De ergonomische resultaten zijn behandeld in het deel voor de informatie-analist. De technische resultaten vallen uiteen in verkeers-, verwerkings- en printresultaten. Verkeersresultaten worden behandeld in het hoofdstuk Netwerkontwerp, de printresultaten zijn deels besproken bij de behandeling van de printparameters en worden verder uitgewerkt in het hoofdstuk Netwerkontwerp. De verwerkingsresultaten vallen uiteen in resultaten ten aanzien van responsetijden en systeembelasting. Beide worden in deze paragraaf besproken.

53.2 Subtransakties en kombitransakties.
Bij komplexe transakties kan de informatie-analist besluiten transakties te splitsen in subtransakties. De transaktie-analist moet de cijfers weer samenvoegen via kombitransakties.

53.3 Relatie met andere projektaktiviteiten.
De transaktie-analist voert zijn werk uit als onderdeel van de projektaanpak. De relatie met andere projektaktiviteiten moet goed worden gehandhaafd in de diverse omgevingen. Deze paragraaf geeft per gedefinieerde omgeving in grote lijnen de werkwijze aan. Details zijn te vinden in de paragrafen over Transaktie analyse.

53.4 Transaktie analyse in distributieve omgevingen.
Ook bij koppelingen tussen twee computersystemen ten dienste van interaktieve toepassingen levert Transaktie analyse een belangrijke bijdrage in de bepaling van de hoeveelheid verkeer. Als voorbeelden dienen een micronetwerk met een fileserver en een koppeling.

53.5 Transakties als entiteitstype.
Transakties dienen in de systeemdokumentatie of in de data dictionary te zijn opgenomen met hun attributen. In deze paragraaf worden de technische attributen besproken. Deze attributen zijn direkt of indirekt resultaat van Transaktie analyse.

53.6 Andere terminals dan beeldschermen.
Hoewel bij interaktieve toepassingen meestal beeldschermen worden gebruikt komen ook andere soorten terminals voor. In deze paragraaf worden die behandeld voorzover ze voor het toepassen van Transaktie analyse van belang zijn.

53.7 Transaktie analyse en Datanet 1.
Bij gebruik van telekommunikatielijnen verdiept niemand zich in de hoeveelheid verkeer uit overwegingen van kosten: men betaalt een vast bedrag per maand. Bij Datanet 1 spelen echter de volume kosten een rol. In deze paragraaf wordt aangegeven dat de resultaten van Transaktie analyse gemakkelijk vertaald kunnen worden naar de segmenten waarmee de PTT rekent bij het opstellen van de rekening.

53.8 Voorbeelden van konklusies.
Het belangrijkste van Transaktie analyse is het vertalen van resultaten van het rekenproces naar konklusies. In aansluiting op de vertaling van ergonomische resultaten naar ergonomische konklusies in het deel voor de informatie-analisten, worden in deze

paragraaf technische resultaten vertaald naar technische konklusies ten aanzien van de konfiguratie en het netwerk.

54.1 Netwerken en geografie.
In het deel van de informatie-analisten is aangegeven hoe de geografie van een bedrijf in kaart gebracht moet worden. Op het laagste niveau moeten daar de werkplekken voorkomen waarvoor transakties zijn ontworpen. De verkeersresultaten van Transaktie analyse maken het ontwerp van het netwerk mogelijk waarmee de werkplekken worden gekoppeld aan computersystemen.

54.2 Informatieplan en netwerkontwerp.
Soms moet er in een zeer vroeg stadium, ver voor de projektselektie, iets gedaan worden aan netwerkontwerp, om een indikatie van de kosten te kunnen geven. Afhankelijk van de tijd en de energie die men erin wil stoppen ontstaan kostenplaatjes van globaal tot redelijk nauwkeurig, als men aanneemt dat de situatie niet of nauwelijks zal veranderen.

54.3 Netwerkontwerp en verkeer.
In de vakliteratuur wordt nog regelmatig geschreven over het ontwerpen van netwerken. Het is niet de bedoeling in herhaling te treden. Transaktie analyse levert de gegevens om het verkeer te bepalen dat de basis vormt onder het netwerkontwerp. In deze paragraaf wordt aangegeven in welke stappen de hoeveelheid verkeer en de bezetting van lijnen wordt berekend.

54.4 Netwerkontwerp in een C1N/C3N-omgeving.
De meest voorkomende netwerkstruktuur is de ster- of de boomstruktuur. Er worden een aantal mogelijkheden besproken om deze netwerken te realiseren. De resultaten van Transaktie analyse zijn de cijfers die de leveranciers van netwerkapparatuur nodig hebben om de alternatieven door te rekenen.

54.5 Netwerkontwerp in distributieve omgevingen.
Distributie van gegevens en het netwerkontwerp beinvloeden elkaar. Soms zijn voor beide nogal wat mogelijkheden. Dat maakt het aantal kombinaties onoverzichtelijk. Bovendien is de evaluatie van de verschillende konstrukties een moeizaam proces. In deze paragraaf wordt de aanpak behandeld.

54.6 Netwerkontwerp en uitwijk.
Uitgangspunt bij het netwerkontwerp ten dienste van computeruitwijk is dat de werkplekken van de gebruikers in takt zijn. De apparatuur op de werkplekken moet door omschakeling verbonden worden met de uitwijkcomputer. Bij het ontwerpen van netwerken moet

de uitwijk vanaf het begin in rekening worden gebracht.

55.1 Responsetijden.
In deze paragraaf worden alle aspekten van responsetijden behandeld, vanaf het stellen van eisen tot aan de evaluatie van gerealiseerde responsetijden.

55.2 Evaluatie van interaktieve toepassingen.
Naast de beoordeling van het koncept en de ergonomische aspekten, maakt een onderzoek naar de performance een belangrijk deel uit van de evaluatie. Slechte performance kan worden veroorzaakt door een slecht ontwerp, gebrekkige optimalisatie of overbelasting. Het gaat daarbij om het database-, programma- en netwerkontwerp.

Deel 6, voor gebruikers.

61.1 Interaktieve toepassingen.
In deze paragraaf worden de termen besproken die in de volgende paragrafen gebruikt worden.

61.2 Vakmanschap.
Automatiseerders bouwen een gereedschap voor de gebruikers. Die gebruikers kunnen vanuit hun vakmanschap het beste bepalen wat er met dat gereedschap gedaan moet worden. Het vakmanschap van de automatiseerders is nodig om de eisen van de gebruikers te koppelen aan de technische mogelijkheden. In feite gaat het om de koppeling van het vakmanschap van de gebruikers aan dat van de automatiseerder.

61.3 Methoden en middelen.
De enige manier voor gebruikers om vat te krijgen op de automatisering is de toepassing van afgesproken methoden. Methoden moeten worden gekozen op basis van hun resultaten voor zowel gebruikers, als voor automatiseerders.

61.4 Projektaanpak.
Meestal hebben de gebruikers weinig inzicht in de manier van werken van de automatiseerders. Toch zitten er in de projektaanpak kritische momenten die voor de gebruikers van belang zijn. Het is voor gebruikers van belang zich aan te sluiten bij de indeling in fasen van de automatiseerders, om zo op het juiste moment de goede vragen te kunnen stellen.

61.5 Kommunikatie met de automatiseerder.
De te behandelen methoden hebben betrekking op de kommunikatie tussen twee vakgebieden. Aan gebruikerszijde is niet altijd even

duidelijk wie optreedt als gesprekspartner, welke gegevens moeten worden verstrekt, etc.

61.6 Zelfdiscipline.
Een groot probleem voor de automatiseerders vormen gebruikers die tijdens de analyse geen tijd hebben, tijdens het ontwerp geen belangstelling hebben en tijdens de bouw ineens met voorstellen komen. Het logisch ontwerp is de fase waarin, voor de gebruikers, alles gebeurt. Hier komt de projektaanpak weer ter sprake.

62.1 Wat zijn transakties?
In deze paragraaf worden de begrippen, besproken in de paragraaf Interaktieve toepassingen, bekend verondersteld. Er wordt vastgesteld wat een transaktie is. Het verband met de schermlay-out wordt besproken. Een handmatige procedure kan tot een of meer transakties leiden. Een belangrijke rol speelt het feit hoe de gebruiker iets ervaart. Het verschil tussen analyse en ontwerp wordt aangegeven.

62.2 Het ontwerpen van transakties.
Van procedures naar transakties, het iteratieve ontwerpproces, de relatie met het logisch en technisch ontwerp, het doel van dialoogsimulatie en Transaktie analyse.

62.3 Cijfers gevraagd.
Cijfers zijn noodzakelijk om de gevolgen van de automatisering voor de gebruikers te berekenen, zowel in ergonomisch als in financieel opzicht. De resultaten van de berekening maken een gevoeligheidsanalyse mogelijk in het geval van moeilijk te schatten cijfers. Bedrijfspolitieke problemen worden niet met cijfers opgelost.

63.1 Dialoogsimulatie als methode.
In een vorige paragraaf zijn methoden in het algemeen besproken. Een vaste manier van werken die is gekozen op basis van resultaten, is voor de gebruiker de enige manier om vat te krijgen op de automatisering. Dit geldt in het bijzonder van methoden voor kommunikatie met automatiseerders. Stapsgewijs wordt de methode dialoogsimulatie besproken. De verschillen met prototyping worden kort aangegeven.

63.2 De dialoogsimulator.
In deze paragraaf wordt het middel bij de methode van de vorige paragraaf besproken. De mogelijkheden en beperkingen van een simulator worden behandeld. De bediening is behandeld in het deel voor de informatie-analist.

63.3 Responsetijden.
Responsetijden zijn de nachtmerries voor de automatiseerders. Het beheer van responsetijden begint hij specifieke en realistische eisen van gebruikers en het verstrekken van juiste cijfers aan informatie-analisten. De dialoogsimulator is onmisbaar bij het bepalen van realistische eisen.

63.4 Eisen gevraagd.
Het is een probleem op zich, om eisen ten aanzien van de automatisering boven water te krijgen bij gebruikers die niet weten wat een beeldscherm is. Er moet onderscheid gemaakt worden naar gebruikers en het soort eisen dat gevraagd worden. Bij dialoogsimulatie gaat het om eisen ten aanzien van de dialoog, maar toch ...

63.5 De kreatieve gebruiker.
Het gevaar is groot dat de automatisering gebaseerd wordt op de huidige situatie. Toch zijn er vaak gebruikers met een uitstekende visie op nieuwe mogelijkheden die computers bieden. Jammer dat die kreativiteit pas gaat werken wanneer het systeem gebouwd is.

64.1 Transaktie analyse.
Het principe van Transaktie analyse wordt beschreven en met name de inbreng van gebruikers en de konsekwenties daarvan. De relatie met de projektaanpak zoals die eerder is besproken komt hier weer naar voren.

64.2 Resultaten van Transaktie analyse.
De twee pagina's output van het rekenprogramma worden behandeld. Elk van de samenstellende delen van de terminaltransaktietijd wordt besproken. Het rekenprogramma doet eigenlijk niet anders dan cijfers van gebruikers omrekenen naar cijfers voor de toekomstige situatie. Konklusies kunnen aanleiding zijn om het ontwerp geheel of gedeeltelijk te herzien.

64.3 Sociale aspekten in cijfers.
Algemene diskussie over de sociale aspekten van automatisering lossen de vragen van een gebruiker op zijn werkplek niet op. Konkreet worden de sociale aspekten pas met cijfers. Tijdens dialoogsimulatie ervaart hij hoe z'n toekomstige werk eruit gaat zien, Transaktie analyse levert de cijfers. Er kan worden gerekend aan: normale werkdagen, piekdagen, overwerkuren, de hoeveelheid overwerk, enzovoort.

64.4 Vergissingen van gebruiker.
In veel bedrijven hebben gebruikers tot hun schade ondervonden dat ze zich niet voldoende bemoeid hebben met de automatisering.

Bij evaluatie van interaktieve systemen blijkt meestal dat op het kritieke moment gebruikers niet aanwezig waren. De argumenten die toen steekhoudend leken, blijken achteraf vergissingen te zijn. Zo is de geautomatiseerde situatie ontstaan voor hen, bij hen en zonder hen.

Lijst van figuren

Verklarende woordenlijst

3270-achtig beeldscherm: Een beeldscherm met een intelligentie en een besturing die overeenkomt met 3270 beeldschermen van IBM.

ACK: Een besturingsteken dat verzonden wordt als bevestiging van een ontvangen bericht (Acknowledge).

Af en toe gebruiker: Een eindgebruiker die weinig achter een terminal zit. Kent geen gebruikersvoorschriften uit zijn hoofd. Wordt nooit geroutineerd.

Afdeling: In het kader van het onderwerp geografie is het een onderdeel van de geografische eenheid: gebouw. Afdelingen kunnen gebruik maken van een LAN voor hun onderlinge kommunikatie.

Afsluitresponse: Interaktie waarvan de responsetijd niet kritisch is voor het werkritme, bijvoorbeeld omdat deze een afsluiting vormt van een reeks kritische interakties en/of het begin vormt van een reeks menselijke handelingen.

Aktiviteiten: Algemeen begrip, maar ook onderdeel van een funktie. Aktiviteiten zijn vaak onder te verdelen tot het niveau van procedures.

Attributen: Eigenschappen van een entiteitstype.

Automatiseerder: Verzamelnaam voor alle funktionarissen van de automatiseringsafdeling die betrokken zijn bij de ontwikkeling, de bouw en het beheer van informatiesystemen: projektadviseurs, informatie-analisten, systeemontwerpers, transak-

tie-analisten, systeemspecialisten, applikatieprogrammeurs etc.

Automatiseerder: Automatiseringsdeskundige. Algemene term, vaak voorkomend in de gebruikers wereld, waar men niet zoveel inzicht heeft in alle automatiseringsfunkties.

Automatiseringsafdeling: Algemene aanduiding voor de funktionele eenheid in de organisatie waartoe alle automatiseerders behoren.

Automatiseringsmanagement: Managers verantwoordelijk voor de automatisering of een deel ervan.

BPS: Het aantal bits dat per seconde getransporteerd wordt, op welke manier dan ook.

Batch-tijdperk: In het kader van dit boek, de periode waarin batch-programma's werden ontwikkeld en de inspraak van de gebruiker beperkt bleef tot de lay-out van de geprinte overzichten.

Batch-verkeer (tussen computersystemen):
- Het oversturen van bestanden.
- Het oversturen van een of meer records a-synchroon met de transaktie die het transport start. Batch-verkeer wordt niet met Transaktie analyse in kaart gebracht.

Baud: Bemonsteringssnelheid. Het aantal keren per seconde dat een ontvangend modem naar de lijn luistert.

Bedrijfsfunktie: Funktie binnen organisatie volgens Fayol (13 pag. 219).

Beeldscherm: Display eenheid (T.V.-scherm) met toetsenbord of een T.V.-scherm zelf.

C1: Omgeving met een centraal rekencentrum.

C1N: Omgeving met een centraal rekencentrum en sternetwerk voor remote terminals.

C2: Omgeving met meerdere rekencentra.

C2N: Omgeving met meerdere gekoppelde rekencentra.

C3: Omgeving met een minicomputer.

C3N: Omgeving met een minicomputer en een netwerk voor remote terminals.

C4: Omgeving met een aantal micro's.

C4N: Omgeving met een aantal gekoppelde micro's.

C5: Een kombinatie van mainframe, mini's en micro's maar zonder koppelingen.

C5N: Een kombinatie van mainframe, mini's en micro's, onderling gekoppeld.

Centraal transaktieschema: Zie transaktieschema.

Combitransaktie: Parameteroverzichten van transakties worden samengevoegd tot een gewogen gemiddelde per werkplek.

Computer: Centrale verwerkingseenheid met I/O poorten. Voor de gebruiker begint de computer direkt achter het T.V.-scherm en

het toetsenbord.

Computerlokatie: Geografisch bepaalde plaats waar een computer is opgesteld.

Computernetwerk: Verzamelnaam voor het geheel van met elkaar verbonden comuters en terminals.

Computersysteem: Computer met peripherie en software, de werkende konfiguratie.

Conversatiediagram: Zie dialoogstruktuurdiagram,

Cx: C1 of C2 of C3 of C4 of C5

CxN: C1N of C2N of C3N of C4N of C5N

Data-analist: Informatie-analist die zich bezig houdt met het analyseren en in kaart brengen van gegevens in de vorm van gegevensmodellen.

Decentraal transaktieschema: Het normale, centrale transaktieschema is aan de rechterzijde uitgebreid met twee kolommen om het verkeer met en de verwerking op een andere lokatie aan te kunnen geven.

Decentrale verwerking: Verwerking op een computersysteem dat op bestandsniveau gekoppeld is met een centraal systeem. Applikaties op beide systemen werken onafhankelijk van elkaar, wat de tijd betreft.

Definitief transaktieschema: Het transaktieschema dat beschikbaar is na de dialoogsimulatie. Definitief is zeer betrekkelijk in het kader van iteraties. Definitief betekent meestal dat het schema geschikt is voor het maken van een detailschema.

Detailschema: Het gekwantificeerde transaktieschema op basis van standaardparameters, kansfaktoren, gemiddelde waarden en varianties.

Dialoog: Het geheel van een of meer interakties tussen mens en computer binnen een transaktie, onafhankelijk van het aantal schermen (schermlay-outs) dat daarbij is betrokken.

Dialoogresponse: Interaktie waarvan de responsetijd kritisch is voor het werkritme.

Dialoogstruktuurdiagram: Exakte en volledige vastlegging van de schermvolgorde: de struktuur en de flow zijn meestal beide aangegeven.

Direkt verkeer: In distributieve omgevingen gaat het om verkeer dat nodig is om de dialoog van een transaktie te kunnen afhandelen. Indirekt verkeer is verkeer dat achteraf nodig is om b.v. gegevens te wijzigen, omdat de afloop van de transaktie dat noodzakelijk maakt.

Displayen: Vertonen op het beeldscherm van gegevens of de schermlay-out.

Distributief systeem: Een systeem dat in staat is gegevens bij een ander systeem op te vragen zonder daarbij afhankelijk te

zijn van zijn plaats in het netwerk, geografisch noch funktioneel.

Distributieve omgeving: Omgeving waar de gegevensopslag geografisch verspreid is om tot een optimaal gegevensgebruik te komen.

Distributieve verwerking: Verwerking op verschillende systemen via gegevensuitwisseling per transaktie. Het verkeer tussen computersystemen maakt deel uit van de responsetijden van de interaktieve toepassingen.

ENQ: Een besturingsteken dat verzonden wordt om te vragen of het andere station in staat is verkeer te ontvangen (Enquiry).

Eindgebruiker: Iemand in een organisatie die werkt met beeldschermen (operator, terminaloperator)

ENTER-toets: Toets waarmee het intypen van een of meer velden wordt afgesloten en de verwerking door het programma wordt gestart. Andere benamingen: CR-, TRANSMIT-, RETURN-, SEND-toets.

Entiteit: Een zaak, ding, persoon, iets konkreets of abstracts waarover in een informatie systeem gegevens zijn opgenomen (9).

Entiteitstype: Een verzameling entiteiten die in de gekozen eigenschappen overeenkomen (9).

Ergonomisch detailschema: Een detailschema dat alleen ergonomische parameters bevat.

Ergonomische parameters: De parameterkodes 1,4,5,7,11,12,13,14 en 16.

Ergonomische resltaten: De resultaten op de pagina terminaltransaktietijd.

Ergonomische Transaktie analyse: Bij deze analyse gaat het om de kwantificering van de menselijke handelingen binnen een transaktie. Verwerkingsaspekten worden in de gewenste responsetijden uitgedrukt.

FEP: Front end processor.

Funktie: Detail van een bedrijfsfunktie, meestal op werkplekniveau.

Funktionaris: Medewerker in een bedrijf, onafhankelijk van zijn plaats in de hierarchie.

Fysieke I/O: I/O op een fysieke database. Of er bij de uitvoering van een fysieke I/O ook gegevens getransporteerd worden via het kanaal hangt af van zaken als, poolsize, blocksize etc.

Gebouw: Geografische eenheid, als onderdeel van een wide area network (WAN) of een lokaal d.c. netwerk.

Gebruikers: Verzamelnaam voor alle medewerkers en funktionarissen die geen deel uitmaken van de automatiseringsafdeling

Gebruikersontwerp: Door gebruikers ontworpen systeemaspekten. Gemaakt in samenwerking met informatie-analisten, in overeen-

stemming met bedrijfsstandaards.

Geheide gebruiker: Een eindgebruiker die dagelijks achter de terminal zit en daardoor geroutineerd ermee om kan gaan.

Globale Transaktie analyse: Het globale slaat op de nauwkeurigheid van de beschikbare gegevens. De dialoog en de beeldscherm lay-out zijn nog niet ontworpen noch de bestanden en db. De verwerkingsaspekten worden globaal aangegeven op het transaktieschema.

Interlokaal: Tussen telefoondistricten van de PTT.

I/O systeem: I/O poort met besturingssoftware.

Indirekt verkeer: Dit verkeer is niet rechtstreeks uit de dialoog aan het beeldscherm af te leiden. Het is nodig om bijvoorbeeld in distributieve omgevingen, na afloop van een transaktie of een serie transakties databases op een andere lokatie te updaten.

Interaktie: In het algemeen: de wisselwerking tussen mens en computer. Hier alleen de aktie die de verwerking door de computer start.

Interaktief batch-verkeer: Verkeer tussen computersystemen synchroon met de transatie en dat bestaat uit het oversturen van een bestand, terwijl de transportvertraging deel uitmaakt van de responsetijd.

Interaktief verkeer (tussen computer systemen): Verkeer dat synchroon loopt met een transaktie. De transporttijd kan deel uitmaken van de responsetijd. Interaktief verkeer wordt op een decentraal transaktieschema aangegeven en verwerkt op het detailschema. Meestal gaat het om korte berichten, enkele records.

Iteratie: Herhaling van een proces met het doel de resultaten te verbeteren.

Kritische interakties: Interakties die deel uitmaken van het werkritme van een gebruiker aan het toetsenbord. Afsluitende interakties behoren daar niet toe.

LAN: Zie local area network.

Leek: Heeft geen flauw idee wat automatiseren is en heeft nog nooit achter een terminal gezeten.

Local area network: Netwerk bestaande uit kabels van een beperkte lengte, die met hoge snelheid gegevens transporteren. Meestal een coax- of een twee-aderige kabel binnen een gebouw of een plant.

Logische I/O: Een fiktieve I/O die uitgevoerd zou moeten worden als het gegevensmodel of de logische database in het geheugen zou zijn opgeslagen zoals het op papier staat. Dus een I/O per genormaliseerd entiteitstype en een per set attributen. Als het gaat om logische I/O's als response-eenheden is er een relatie met de tijd per responsetijd. Logische I/O's

duren 0,1 sec. tenzij er een duidelijke reden is iets anders aan te nemen.

Logische Transaktie analyse: Deze analyse is gelijk aan de ergonomische, alleen is nu een fiktief beeldscherm in het model opgenomen om toch enige indruk te krijgen van de verkeersaspekten en de verwerkingsaspekten op basis van het logisch gegevensontwerp.

Logische Transaktie analyse: Gelijk aan ergonomische analyse, alleen is nu de verwerking op basis van het logisch gegevensontwerp zoveel mogelijk in kaart gebracht

Logisch funktiemodel: Het ontwerp van de benodigde funkties zoals die gerealiseerd zullen worden door het informatie verwerkende systeem, zonder de invloed van een bepaald systeem of een konfiguratie.

Logisch gegevensontwerp: Logisch bestandsontwerp of logisch database-ontwerp, onafhankelijk van een DBMS.

Logisch ontwerp: Resultaten van deze fase zijn: gegevensmodel, funktiemodel. Transaktie ontwerp en dialoog ontwerp is uitgevoerd. Soms genoemd funktioneel ontwerp.

Lokaal: Binnen een telefoondistrict van de PTT.

Lokaal netwerk: Netwerk dat gebruik maakt van lokale PTT lijnen.

Lokale terminals: Terminals die via een V24 kabel, maar zonder PTT lijnen zijn aangesloten op een computer.

Lokatie: Plaats, geografische eenheid.

Lijnprocedure: De vastgelegde kommunikatieprocedure waaraan de stations die zijn aangesloten op een lijn zich moeten houden.

M1: Omgeving waar niet volgens een bepaalde systeemontwikkelingsmethode wordt gewerkt en zonder gestandaardiseerde methoden.

M2: Omgeving waar een aantal methoden algemeen worden toegepast.

M3: Omgeving met een duidelijke systeemontwikkelingsmethode en waar maximaal gebruik gemaakt wordt van de computer als hulpmiddel in het analyse- en ontwerptrajekt.

Masker: Alle vast tekst van een schermlay-out.

Materiedeskundige: Gebruiker die een bedrijfsproces zo volledig kan beschrijven dat automatisering mogelijk wordt.

Methode: Uitgekristalliseerde, gestandaardiseerde en goed gedokumenteerde manier van werken al dan niet met gebruikmaking van middelen, gereedschappen.

Middelen: Gereedschappen, tools.

Multidrop, multipoint: Een datakommunikatie lijn bestuurd door een computersysteem met daaraan verscheidene tributary stations.

Netwerk: Het geheel van verbindingen tussen terminals en computersystemen.

Netwerktopologie: De geografische lay-out van het netwerk.

P1: Het opzetten van automatiseringsplan met een globaal net-

werkontwerp.

P2: Netwerkontwerp op basis van de gegevensdistributie.

P3: Nieuwe toepassingen op bestaande systemen

P4: Nieuwe toepassingen op nieuwe hardware

P5: Nieuwe toepassingen in verband met de overgang van situatie Cx naar CxN.

P6: Overgang van batch-verwerking naar interaktieve toepassingen.

Plant: Bedrijf of bedrijfsonderdeel. Meestal bestaand uit verscheidene gebouwen en met een afgebakend eigen terrein.

Point-to-point: Een verbinding tussen twee systemen die er op contention-basis gebruik van maken.

POV: Procedure overhead. Extra tijd en tekens binnen een lijnprocedure voor een goede besturing van de uitwisseling van berichten.

Printparameters: De parameterkodes 4 en 23.

Proces: Op bedrijfsniveau: het geheel van aktiviteiten uitgevoerd door een of meer funktionele eenheden.
Op computerniveau: het geheel van funkties uitgevoerd door een of meer programma's.

Projektcluster: Een zo groot mogelijke groep projekten waarvoor eerst de fasen tot en met het logisch ontwerp kunnen worden uitgevoerd, om daarvoor de konfiguratie te bepalen of de uitbreiding ervan.

Remote terminals: Terminals die via PTT lijnen zijn aangesloten.

Response tijd: Tijd tussen "ENTER" en eerste teken op het scherm.

Scherm: Schermlay-out, masker.

Schermlay-out: schermindeling.

Strategisch management: Direktie, vestigingsdirekteuren, staffunktionarissen.

Subscherm: Deel van een 24x80-scherm. "Map" in b.v. CICS.

Subtransaktie: Een deel van een transaktie, resulterend in een detailschema.

Systeemontwerpers: Zijn verantwoordelijk voor het technisch ontwerp.

Taktisch management: De managers die verantwoordelijk zijn voor de aanpak om de gestelde bedrijfsdoelen te bereiken.

Technische gegevens: Technisch bestandsontwerp of fysiek database-ontwerp.

Technische parameters: De verkeers-, verwerkings- en printparameters.

Technische Transaktie analyse: Deze analyse is gelijk aan de logische alleen is nu de intelligentie van de gekozen beeldschermen in het model opgenomen en het fysieke database - of het bestandsontwerp. Deze analyse wordt uitgevoerd tijdens

het technisch ontwerp door de detailschema's van de logische Transaktie analyse aan te passen aan technische gegevens van beeldscherm en database en levert nu technisch juiste gegevens op voor het verkeer.

Terminal: Algemene aanduiding voor peripherie apparaat in dienst van de gebruiker b.v. beeldscherm, keyboard-printer, printer.

Terminalemulatie: Een systeem precies zo laten reageren als een bepaalde terminal

Terminalnetwerk: Centrale computer met remote terminals.

Terminal-transaktietijd: Het aantal seconden dat nodig is om een transaktie, zoals die is vastglegd op het transaktieschema, uit te voeren.

Toepassing: Een verzameling van een of meer programma's die een bepaalde funktie realiseert. Voor de gebruikers een of meer transakties.

Topologie: De geografische lay-out.

Topologie van gegevensgebruik: De in kaart gebrachte geografie van de werkplekken waar transakties worden uitgevoerd.

Topologie van gegevensopslag: De in kaart gebrachte geografie van de plaatsen waar computersystemen met bestanden en/of databases zijn opgesteld.

Transaktie: Procedure, werkwijze aan het beeldscherm.

Transaktie-analist: Systeem ontwerper of informatie-analist die een opleiding tot transaktie-analist heeft gehad (17).

Transaktie analyse: Een methode om de procedure aan een beeldscherm kwalitatief en kwantitatief in kaart te brengen, ontworpen door H.L.C.N.M. Suys van Philips Telecommunicatie en Informatie-systemen B.V. te Den Haag. Hij ontwikkelde ook het rekenprogramma dat bij deze methode hoort.

Transaktiedefinitie: De opeenvolging van schermen zoals die gedefinieerd is voor de dialoogsimulatie. Basis voor dialoogstruktuurdiagram, maar hoeft daar nog niet gelijk aan te zijn.

Transaktie detailschema: Zie detailschema.

Transaktie-ontwerp: Het geheel van het opstellen van een transaktieschema, het simuleren van de dialoog op een dialoogsimulator en het kwantificeren van de transaktie via Transaktie analyse.

Transaktie-ontwerper: Informatie-analist met het vakmanschap zoals dat beschreven is in het eerste hoofdstuk van het deel voor de informatie-analist.

Transaktieprotocol: Zie Dialoogstruktuurdiagram.

Transaktieschema: Protocol waarmee een transaktie is vastgelegd.

Tributary station: Een terminal of een computersysteem dat gepollt wordt in een multidrop omgeving.

T.T.T: Zie terminal-transaktietijd

Tijdbestedingsdiagram: Een overzicht dat aangeeft hoeveel tijd per dag per werkplek per transaktie aan een beeldscherm gewerkt wordt.

Verkeer: Aantal tekens dat per tijdseenheid getransporteerd wordt.

Verkeersparameters: De parameterkodes 2,3, en 21.

Verkokering: Specialisatie die samengaat met een zekere blindheid oftewel specialisten op hun eilandjes.

Verwerkingsfunktie: Verwerking door de computer om een bepaald resultaat op te leveren. Meestal de verwerking per interaktie.

Verwerkingsparameters: De parameterkodes 6, 8 en 22.

Vestigingsplaats: Geografische plaatsaanduiding van een bedrijf: een stad, een plaats, een dorp.

Wait and think time: De tijd tussen twee interakties, dus het komplement van de responsetijd.

WAN: Zie Wide area network.

Werkplek: Geografisch bepaalde plaats waar de gebruiker met het beeldscherm werkt.

Werkplekgroep: Groep van werkplekken binnen een organisatorische eenheid of een groep werkplekken waar dezelfde transakties worden uitgevoerd.

Wide area network: Een netwerk waarvan de verbindingen gerealiseerd zijn met PTT voorzieningen als huurlijnen, geschakelde lijnen of Datanet 1.

Trefwoordenlijst

Literatuurlijst

1 Van gegevenverwerking naar kantoorautomatisering.
Een RAET-rapport opgesteld door: B.J. van de Enk.

2 Analyse van transakties in een datakommunikatie omgeving.
Opgesteld door: J.L.C.N.M. Suys. Handboek in (17)

3 Responsetijden: belangrijker dan u verwacht.
Ir. R. Paans en G.E. Houtekamer. Informatie oktober 1983

4 Een systeemontwikkelingsmethode gebaseerd op prototyping.
J. van Beek. Informatie december 1982

5 Ervaringen rond prototyping
Dr. H.G. Sol. Informatie maart 1984

6 Evaluation of methods and techniques for the analysis, design and implementation of information systems.
J. Blank and M.J. Krijger.
Uitgever Academic Service.

7 Information engineering
James Martin en Clive Finkelstein. Savant institute.

8 Systems analysis for data transmission
James Martin

9 Poly automatiseringszakboekje. PBNA 1984

10 Systeem-ontwikkeling volgens SDM
H.B. Eilers. Uitgever Academic Service

11 MiniPARAET. Samenvatting van de PARAET ontwikkelingsmethode voor te automatiseren informatiesystemen.
Uitgave: RAET Arnhem

12 Computersystems in workdesign - the ETICS method.
Enid Mumford and Mary Weir

13 Poly economisch zakboekje. PBNA 1981.

14 Mens-computer-interaktie en technieken voor het beschrijven van dialogen. Informatie, jaargang 26 nr 10 pag. 769-860.

15 Beeldschermgebruiker en dialoogontwerp.
Tweedaagse RAET-kursus voor gebruikers.

16 Dialoogsimulatie en transaktie-ontwerp.
Driedaagse RAET-kursus voor informatie-analisten, systeemontwerpers en programmeurs.

17 Transaktie analyse.
Vierdaagse RAET-kursus voor informatie-analisten, systeemontwerpers, netwerkontwerpers, netwerkbeheerders en hun management.

18 Computer networks.
Andrew S. Tanenbaum. Uitgever Prentice/Hall international.

19 Beeldschermen: Twee uur werken-kwartier rust.
Artikel van J.L.C.N.M. Suys in kantoor en efficiency.
januari 1984

20 ARTIST: Assyst-RAET Tool for Information Systems Technology.
RAET Arnhem

21 Ontwikkelingsmethoden 23. Valkuilen bij automatieprojekten.
Ir. S.M. Argelo. Informatie maart 1982.

22 RAET-dialoogsimulator DS4.x, leverbaar voor de IBM-PC (-XT)

RAET arnhem

23 Ontwikkelingsmethoden 24. Gebruikersparticipatie
J.F. den Hertog en E. van de Wee. Informatie maart 1982
pag. 131-190

24 Projektaanpak Uitwijk. Projektmatige aanpak voor het ontwerpen en beheren van systemen geschikt voor computeruitwijk. Computer Uitwijk Centrum, Lelystad.

25 Datacentre management. Een driedaagse RAET-kursus voor automatiseringsmanagers, hoofden van rekencentra, etc.

26 Computer networks
Andrew S. Tanenbaum. Uitgever Prentice/Hall international

27 Computernetwerken
Dr. C. de Backer. Uitgever Stenfert Kroese.

28 Selection of Data Communications Equipment. Prepared by Management Group Logica LTD, London.

29 Methods for Using Information for Capacity Planning EDP Performance Review november 1984

30 A measure of transactions processingpower
Datamation sept. 1985. Prepared by 24 computer professionals

31 Inleiding datatransmissie
J.H. Balvert en D.J. Elbers. Uitgever Samson, Alphen ad Rijn

32 Data Communications, Facilities, Networks and Systems Design.
Dixon R. Doll.

33 Systeem-ontwikkeling volgens SDM
H.B. Eilers. Uitgever Academic Service

34 Selecting a distributed system.
John C. French. Datamation februari 1981

35 ARTIST een computerondersteund gereedschap bij informatieplanning.
H.A. Huis in't Veld. NGI-SION 1985. Informatica symposium.

36 Inleiding datakommunikatie, Datakommunikatie en netwerken.
Eendaagse, resp. driedaagse RAET-kursus, waarin alle

technische, funktionele en software aspekten van datakommunikatie worden behandeld.

37 Data communicatie
P.C. den Heijer en R. Tolsma. Uitgever Kluwer technische boeken.

38 Centralisation vs decentralisation of information systems
Draft 1977, John F. Rockart.

39 Methoden voor informatiebeleid.
Prof. dr. Th.M.A. Bemelmans en dr. E. Eloranto
Informatie nr. 7 en 8 1983

40 Drs. G.M.V. van Aardenne tijdens een lezing op de konferentie: Data: beheer en kontrole, Noordwijkerhout, mei 1985

41 Interview met prof. Enid Mumford door drs J. van Waerden en drs. A. Lasance in Informatie febr. 1985.

42 De praktijk van informatiebeleid en informatieplanning nader onderzocht.
Dr.ir. E.A. Mantz. Informatie maart 1985

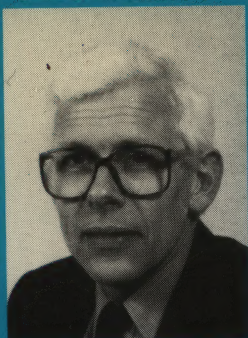
43 Prototyping in een database-omgeving
Drs. F.G. Ras. Informatie maart 1985

44 Statistiek en wachtrijtheorie in een datakommunikatie-omgeving.
Drs. F. Soomers. Syllabus in (17)

45 Data base systemen voor de praktijk
Prof.dr. J.A. Vandenbulcke, Uitgever Kluwer-Antwerpen, Deventer



Over de auteur



De schrijver begon zijn loopbaan in 1956 als electronicus bij Philips Eindhoven. Vervolgens trad hij in dienst bij GEVEKE-GROENPOL in Amsterdam als technisch-commercieel medewerker voor analoge meet- en regelapparatuur in de procesautomatisering. Laatste funktie: chef van een verkoopafdeling.
In 1970 ging hij naar PHILIPS-ELECTROLOGICA in Apeldoorn, als docent hardware en systeemsoftware voor P1000-systemen. Later was hij hoofd software-opleidingen. Vervolgens schakelde hij over naar het vakgebied datakommunikatie. Eerst voor het geven van kursussen en software support, later voor produktplanning.
Sinds 1978 werkt hij bij RAET in Arnhem als consultant voor datakommunikatie, ergonomie en kantoorautomatisering. Hij beschikt over zeven jaar ervaring als adviseur bij het ontwerpen van interaktieve toepassingen en netwerken, het evalueren van bestaande systemen, het ontwikkelen van methoden, tools en kursussen op het gebied van datakommunikatie, ergonomie en kantoorautomatisering en het houden van presentaties voor het topmanagement, automatiseerders en gebruikers.

Over het boek

Mislukkingen en problemen in de automatisering zijn voor de helft te wijten aan automatiseerders en voor de andere helft aan gebruikers. Problemen die gebruikers hebben met de professionele automatisering zijn niet op te lossen met burgerinformatica of softe trainingssessies. Ze moeten als kreatieve mede-ontwerpers worden ingezet bij het ontwerpproces. Daarbij dienen methoden te worden toegepast die passen bij gebruikers en waar automatiseerders verder mee kunnen. In de vakliteratuur over netwerkontwerp komen gebruikers of automatiseerders voor als leveranciers van gegevens die de netwerkontwerpers nodig hebben, maar die ze nooit krijgen. Toch zou het netwerk en het computersysteem gebaseerd moeten zijn op eisen die gebruikers stellen ten aanzien van de funkties die het totale systeem moet verrichten. Dit boek behandelt de kommunikatie tussen gebruikers en automatiseerders en als afgeleide daarvan de kommunikatie tussen automatiseerders en netwerkontwerpers.
Daarbij ervaart de gebruiker tijdens het ontwerp zijn uiteindelijke situatie: kwalitatief, wat betekent het beeldscherm voor mijn werk en kwantitatief, hoe veel uren per dag werk ik ermee, wat gebeurt er in pieksituaties?
De evaluatie van interaktieve toepassingen heeft plaats tijdens het ontwerp, zelfs als de computer nog niet is aangeschaft!